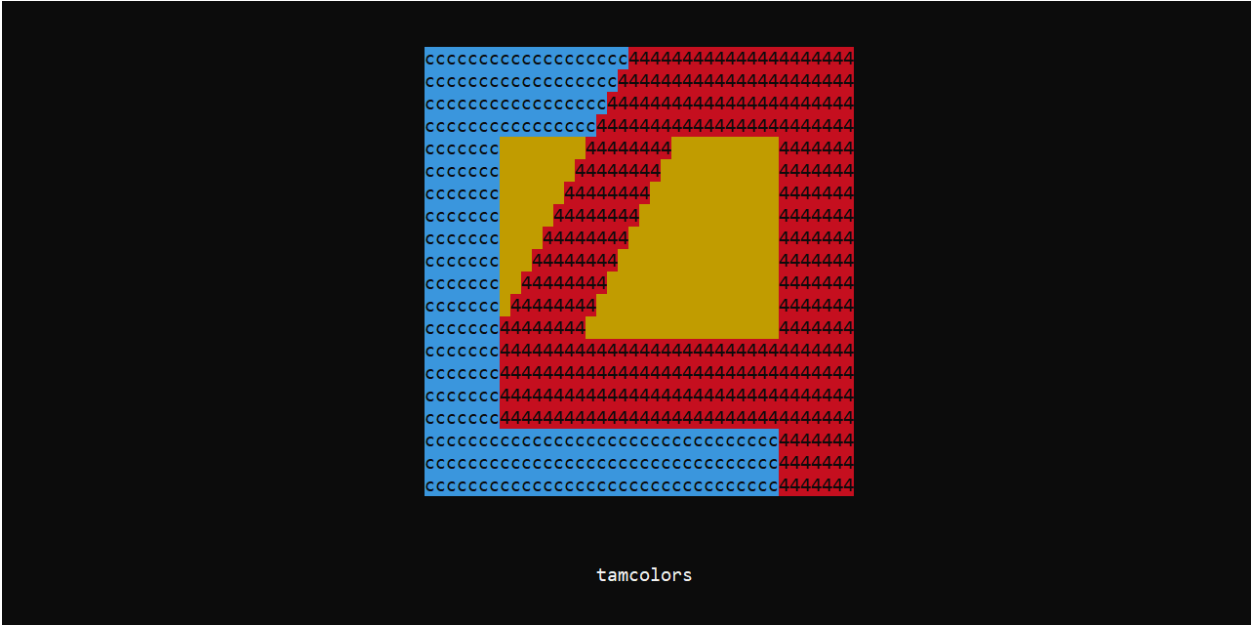

tamcolors

Release 1.2.0

Nov 03, 2020

Contents:

1	about	3
2	pip install tamcolors	5
3	links	7
4	table tennis example	9
5	basic example	11
6	icon example	13
6.1	tamcolors	13
7	Indices and tables	61
	Python Module Index	63
	Index	65



CHAPTER 1

about

This library standardizes console color output across multiple platforms. It also can get keyboard inputs without interrupting the console output and can run a console application at ~25 FPS.

CHAPTER 2

`pip install tamcolors`

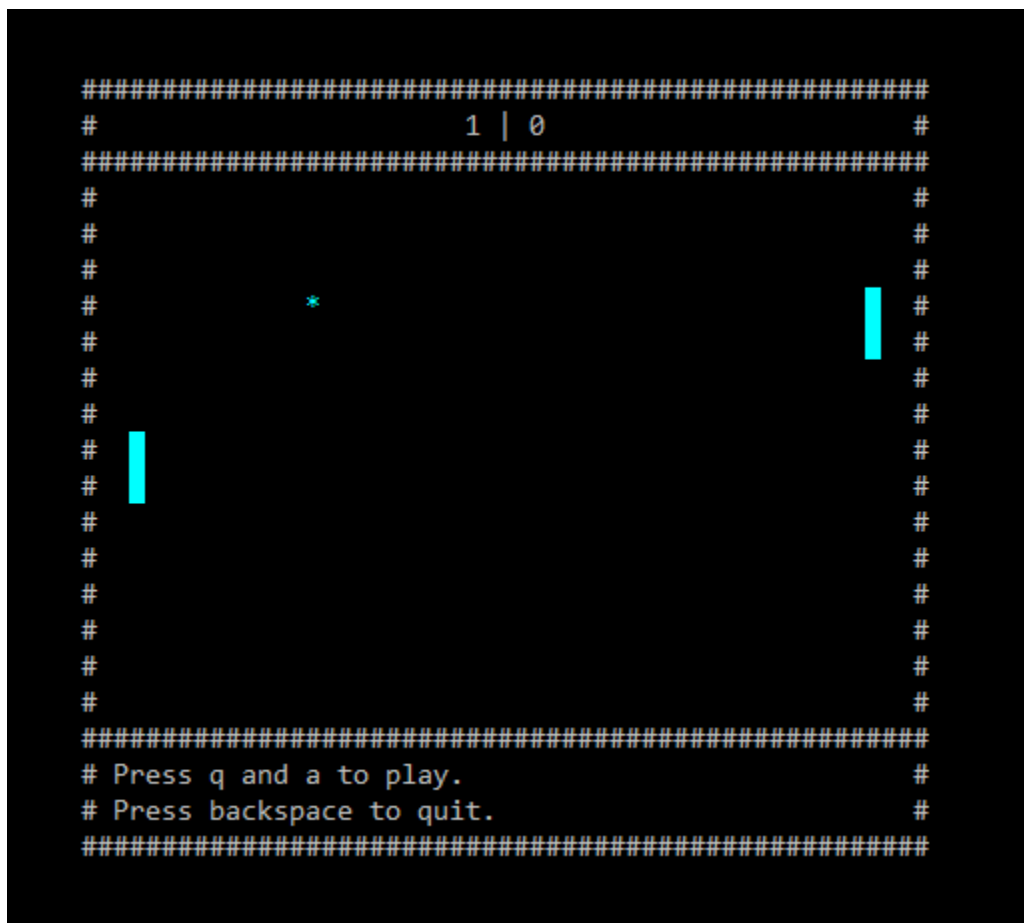
CHAPTER 3

links

- [github](#)
- [pypi](#)
- [read the docs](#)
- [youtube](#)

CHAPTER 4

table tennis example



```
import tamcolors
tamcolors.examples.tabletennis.run()
```


CHAPTER 5

basic example

```
>>> tamcolors.examples.basic.run()  
Hello World!  
Whats Your Name? >>> Chad
```

```
import tamcolors  
tamcolors.examples.basic.run()
```

```
import tamcolors  
from tamcolors.tam_io.tam_colors import *  
tamcolors.printc("Hello", "World!", ("light blue", "white"), same_color=True)  
name = tamcolors.inputc("Whats Your Name? >>> ", ("light aqua", "gray"))  
tamcolors.clear()  
tamcolors.printc("Hello, ", ("default", "default"), name, (GREEN, WHITE), "!", ("gray  
↪", "light aqua"), sep="")
```


CHAPTER 6

icon example

```
import tamcolors
tamcolors.examples.icon.run()
```

6.1 tamcolors

6.1.1 tamcolors package

Subpackages

tamcolors.examples package

Submodules

tamcolors.examples.alpha module

```
class tamcolors.examples.alpha.TAMAlpha
    Bases: tamcolors.tam.tam_loop.TAMFrame
    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:
    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:
tamcolors.examples.alpha.run()
```

tamcolors.examples.basic module

```
tamcolors.examples.basic.run()
```

tamcolors.examples.clouds module

```
class tamcolors.examples.clouds.Clouds
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.clouds.run()
```

tamcolors.examples.colors module

```
class tamcolors.examples.colors.TAMCOLORS
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.colors.run()
```

tamcolors.examples.connection_loopback module

```
tamcolors.examples.connection_loopback.run()
```

tamcolors.examples.host_loopback module

```
tamcolors.examples.host_loopback.run()
```

tamcolors.examples.icon module

```
class tamcolors.examples.icon.BootLogo
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.icon.run()
```

tamcolors.examples.rgb_color module

```

class tamcolors.examples.rgb_color.RGBCOLOR
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.rgb_color.run()

```

tamcolors.examples.sandy_check module

```

tamcolors.examples.sandy_check.run()

```

tamcolors.examples.tabletennis module

```

class tamcolors.examples.tabletennis.Ball (tam_buffer, way=False)
    Bases: object

    get_y ()

    update ()

    winner ()

class tamcolors.examples.tabletennis.Racket (x, ball, tam_buffer, ai=False)
    Bases: object

    update (key_manager)

class tamcolors.examples.tabletennis.TableTennis
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.tabletennis.run()

```

tamcolors.examples.tam_key_manager module

```

class tamcolors.examples.tam_key_manager.TAMKeyManager
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

```

```
tamcolors.examples.tam_key_manager.run()
```

tamcolors.examples.tam_list_buffer module

```
class tamcolors.examples.tam_list_buffer.TAMListBuffer
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.tam_list_buffer.run()
```

tamcolors.examples.tam_loop module

```
class tamcolors.examples.tam_loop.TAMLoopHelloWorld
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.tam_loop.run()
```

tamcolors.examples.tam_print module

```
class tamcolors.examples.tam_print.TAMPrint
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:

tamcolors.examples.tam_print.run()
```

tamcolors.examples.tam_text_box module

```
class tamcolors.examples.tam_text_box.TAMPrint
    Bases: tamcolors.tam.tam_loop.TAMFrame

    draw (tam_buffer, loop_data)
        info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

    update (tam_loop, keys, loop_data)
        info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict
        :return:
```

```
tamcolors.examples.tam_text_box.run()
```

Module contents

tamcolors.tam package

Submodules

tamcolors.tam.tam_loop module

```
class tamcolors.tam.tam_loop.TAMFrame(fps, char, foreground_color, background_color,  
                                     min_width=0, max_width=1000, min_height=0,  
                                     max_height=1000)
```

Bases: object

done (*tam_loop, loop_data*)
info: will clean up the frame and can only be called once :param tam_loop: TAMLoop :param loop_data: dict :return:

draw (*tam_buffer, loop_data*)
info: will draw frame onto terminal :param tam_buffer: TAMBuffer :param loop_data: dict :return:

get_defaults ()
info: gets defaults :return: (str, int, int)

get_fps ()
info: returns the frame fps :return: int

get_height_min_and_max ()
info: returns min and max height :return: (int, int)

get_width_min_and_max ()
info: returns min and max width :return: (int, int)

make_buffer_ready (*tam_buffer, screen_width, screen_height*)
info: will make buffer ready for frame :param tam_buffer: TAMBuffer :param screen_width: int: 0 - inf
:param screen_height: int: 0 - inf :return:

update (*tam_loop, keys, loop_data*)
info: will update terminal :param tam_loop: TAMLoop :param keys: list, tuple :param loop_data: dict :return:

```
class tamcolors.tam.tam_loop.TAMLoop(tam_frame,          io=None,          only_any_os=False,  
                                   color_change_key='ESCAPE',      loop_data=None,  
                                   stability_check=False,      tam_color_defaults=True,  
                                   highest_mode_lock=False,    preferred_mode=None,  
                                   test_mode=False)
```

Bases: object

add_frame_stack (*frame*)
info: will add a TAMFrame to stack :param frame: TAMFrame :return:

done (*reset_colors_to_console_defaults=True*)
info: will stop tam loop :param: reset_colors_to_console_defaults: bool :return: None

get_color_16 (*spot*)
info: Will get color from color palette 16 :param spot: int :return: RGBA

get_color_16_pal_256 (*spot*)
info: Will get color from color palette 2 :param spot: int :return: RGBA

get_color_2 (*spot*)
info: Will get color from color palette 2 :param spot: int :return: RGBA

get_color_256 (*spot*)
info: Will get color from color palette 256 :param spot: int :return: RGBA

get_running ()
info: None has not ran, True is running, False has ran :return: bool or None

pop_frame_stack ()
info: will remove TAMFrame from stack :return: TAMFrame or None

reset_colors_to_console_defaults ()
info: will reset colors to console defaults :return: None

run ()
info: will call tam loop :return: None

static run_application (*args, **kwargs)
info: will run tam loop as an application note: when tam loop is done running the program will quit if tam loop has an error and the frame does not catch it the error will be printed onto the screen and the program will quit after user input :param args: :param kwargs: :return:

run_with_profiler ()
info: will run with a profiler and print out data when done :return: None

set_color_16 (*spot, color*)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_16_pal_256 (*spot, color*)
info: sets a color value :param spot: int :param color: int :return: None

set_color_2 (*spot, color*)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_256 (*spot, color*)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_tam_color_defaults ()
info: will set console colors to tam defaults :return: None

step (*keys=()*)
info: only works in test mode but lets you update and draw in tests :param keys: tuple: ((str, str), ...)
:return: TAMBuffer or None

exception tamcolors.tam.tam_loop.TAMLoopError
Bases: Exception

Module contents

tamcolors.tam_basic package

Submodules

tamcolors.tam_basic.basic module

`tamcolors.tam_basic.basic.clear()`

info: Clears the console :return: None

`tamcolors.tam_basic.basic.inputc(output, color)`

info: Will get color input from console :param output: str :param color: tuple :return: str

`tamcolors.tam_basic.basic.printc(*output, same_color=False, sep=' ', end='\n', flush=True, stderr=False)`

info: Prints color output to the console :param output: :param same_color: bool :param sep: str :param end: str :param flush: bool :param stderr: std :return: None

`tamcolors.tam_basic.basic.reset_colors_to_console_defaults()`

info: will reset colors to console defaults :return: None

`tamcolors.tam_basic.basic.set_tam_color_defaults()`

info: will set console colors to tam defaults :return: None

Module contents

tamcolors.tam_c package

Module contents

tamcolors.tam_io package

Submodules

tamcolors.tam_io.ansi_256_drivers module

class `tamcolors.tam_io.ansi_256_drivers.ANSI256ChangerDriver(*args, **kwargs)`

Bases: `tamcolors.tam_io.tam_drivers.ColorChangerDriver`, `abc.ABC`

class `tamcolors.tam_io.ansi_256_drivers.ANSI256ColorDriver(*args, **kwargs)`

Bases: `tamcolors.tam_io.tam_drivers.FullColorDriver`, `abc.ABC`

draw (`tam_buffer`)

info: Will draw TAMBuffer to console :param tam_buffer: TAMBuffer :return: None

get_printc_mode ()

Gets the modes used by printc and inputc :return: str

inputc (`output, color`)

info: Will get input from the console in color :param output: str :param color: COLOR :return: str

printc (`output, color, flush, stderr`)

info: Will print to the console in color :param output: str :param color: COLOR :param flush: bool :param stderr: std :return:

set_mode (`mode`)

info: will set the color mode :param mode: int: key to color mode :return:

start ()

info: operations for IO to start :return: None

tamcolors.tam_io.ansi_true_color_drivers module

```
class tamcolors.tam_io.ansi_true_color_drivers.ANSITrueFullColorDriver (*args,  
                                                                    **kwargs)  
    Bases: tamcolors.tam_io.tam_drivers.FullColorDriver, abc.ABC  
  
    draw (tam_buffer)  
        info: Will draw TAMBuffer to console :param tam_buffer: TAMBuffer :return: None  
  
    get_printc_mode ()  
        Gets the modes used by printc and inputc :return: str  
  
    inputc (output, color)  
        info: Will get input from the console in color :param output: str :param color: COLOR :return: str  
  
    printc (output, color, flush, stderr)  
        info: Will print to the console in color :param output: str :param color: COLOR :param flush: bool :param  
        stderr: std :return:  
  
    set_mode (mode)  
        info: will set the color mode :param mode: int: key to color mode :return:  
  
    start ()  
        info: operations for IO to start :return: None
```

tamcolors.tam_io.any_drivers module

```
class tamcolors.tam_io.any_drivers.ANYColorChangerDriver (*args, **kwargs)  
    Bases: tamcolors.tam_io.tam_drivers.ColorChangerDriver, abc.ABC  
  
class tamcolors.tam_io.any_drivers.ANYColorDriver (*args, **kwargs)  
    Bases: tamcolors.tam_io.tam_drivers.ColorDriver, abc.ABC  
  
    draw (tam_buffer)  
        info: Will draw TAMBuffer to console :param tam_buffer: TAMBuffer :return: None  
  
    get_printc_mode ()  
        Gets the modes used by printc and inputc :return: str  
  
    inputc (value, color)  
        info: will get user input with color :param value: str :param color: tuple: (int, int) :return: str  
  
    printc (output, color, flush, stderr)  
        info: will print out user output with color :param output: str :param color: tuple: (int, int) :param flush:  
        boolean :param stderr: boolean :return: None  
  
class tamcolors.tam_io.any_drivers.ANYKeyDriver (*args, **kwargs)  
    Bases: tamcolors.tam_io.tam_drivers.KeyDriver, abc.ABC  
  
    get_key ()  
        info: Gets an input from the terminal :return: tuple or false  
  
    static get_key_dict ()  
        info: Gets a dict of all the keys :return: {str: (str, str), ... }  
  
class tamcolors.tam_io.any_drivers.ANYUtilitiesDriver (*args, **kwargs)  
    Bases: tamcolors.tam_io.tam_drivers.UtilitiesDriver, abc.ABC  
  
    clear ()  
        info: Will clear the console :return: None
```


get_dimensions()
 info: Gets the dimensions of console :return: (int, int): (row, column)

show_console_cursor(show)
 info: Will show or hide console cursor :param show: int :return: None

tamcolors.tam_io.io_tam module

```
class tamcolors.tam_io.io_tam.IO(identifier,      mode_2=True,      mode_16_pal_256=True,
                                mode_16=True,      mode_256=True,
                                mode_rgb=True,      key_driver_operational=True,
                                color_driver_operational=True,
                                color_changer_driver_operational=True,      utili-
                                ties_driver_operational=True)
```

Bases: *tamcolors.tam_io.io_tam.RawIO*, *abc.ABC*

classmethod able_to_execute()
 info: checks that io is stable in current environment :return: bool

clear()
 info: Will clear the console :return: None

color_changer_driver_operational()
 info: checks if the color changer driver is operational :return: bool

color_driver_operational()
 info: checks if the color driver is operational :return: bool

done()
 info: operations for IO to stop :return: None

draw(tam_buffer)
 info: Will draw TAMBuffer to console :param tam_buffer: TAMBuffer :return: None

enable_console_keys(enable)
 info: will enable console keys :param enable: boool :return: None

get_color_16(spot)
 info: Will get color from color palette 16 :param spot: int :return: RGBA

get_color_16_pal_256(spot)
 info: Will get color from color palette 2 :param spot: int :return: int

get_color_2(spot)
 info: Will get color from color palette 2 :param spot: int :return: RGBA

get_color_256(spot)
 info: Will get color from color palette 256 :param spot: int :return: RGBA

get_dimensions()
 info: Gets the dimensions of console :return: (int, int): (row, column)

get_info_dict()
 info: will get the identifier dict :return: dict

get_key()
 info: Gets an input from the terminal :return: tuple or false

static get_key_dict()
 info: Gets a dict of all the keys :return: {str: (str, str), ... }

get_mode()
info: will return the current color mode :return: int

get_modes()
info: will return a tuple of all color modes :return: (int, int, ...)

get_printc_mode()
Gets the modes used by printc and inputc :return: str

inputc(output, color)
info: Will get input from the console in color :param output: str :param color: COLOR :return: str

is_console_cursor_enabled()
info: will check if console cursor is enabled :return: bool

is_console_keys_enabled()
info: will check if console keys enabled :return: bool

key_driver_operational()
info: checks if the key driver is operational :return: bool

printc(output, color, flush, stderr)
info: Will print to the console in color :param output: str :param color: COLOR :param flush: bool :param stderr: std :return:

reset_colors_to_console_defaults()
info: will reset colors to console defaults :return: None

set_color_16(spot, color)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_16_pal_256(spot, color)
info: sets a color value :param spot: int :param color: int :return: None

set_color_2(spot, color)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_256(spot, color)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_mode(mode)
info: will set the color mode :param mode: int: key to color mode :return: None

set_tam_color_defaults()
info: will set console colors to tam defaults :return: None

show_console_cursor(show)
info: Will show or hide console cursor :param show: bool :return: None

start()
info: operations for IO to start :return: None

utilities_driver_operational()
info: checks if the utilities driver is operational :return: bool

class tamcolors.tam_io.io_tam.RawIO
Bases: abc.ABC

classmethod able_to_execute()
info: checks that io is stable in current environment :return: bool

clear()
info: Will clear the console :return: None

color_change_driver_operational ()
 info: checks if the color changer driver is operational :return: bool

color_changer_driver_operational ()
 info: checks if the color changer driver is operational :return: bool

color_driver_operational ()
 info: checks if the color driver is operational :return: bool

done ()
 info: operations for IO to stop :return: None

draw (*tam_buffer*)
 info: Will draw TAMBuffer to console :param tam_buffer: TAMBuffer :return: None

enable_console_keys (*enable*)
 info: will enable console keys :param enable: boool :return: None

get_color_16 (*spot*)
 info: Will get color from color palette 16 :param spot: int :return: RGBA

get_color_16_pal_256 (*spot*)
 info: Will get color from color palette 2 :param spot: int :return: int

get_color_2 (*spot*)
 info: Will get color from color palette 2 :param spot: int :return: RGBA

get_color_256 (*spot*)
 info: Will get color from color palette 256 :param spot: int :return: RGBA

get_dimensions ()
 info: Gets the dimensions of console :return: (int, int): (row, column)

get_info_dict ()
 info: will get the identifier dict :return: dict

get_key ()
 info: Gets an input from the terminal :return: tuple or false

static get_key_dict ()
 info: Gets a dict of all the keys :return: {str: (str, str), ... }

get_mode ()
 info: will return the current color mode :return: int

get_modes ()
 info: will return a tuple of all color modes :return: (str, str, ...)

get_printc_mode ()
 Gets the modes used by printc and inputc :return: str

inputc (*output, color*)
 info: Will get input from the console in color :param output: str :param color: COLOR :return: str

is_console_cursor_enabled ()
 info: will check if console cursor is enabled :return: bool

is_console_keys_enabled ()
 info: will check if console keys enabled :return: bool

key_driver_operational ()
 info: checks if the key driver is operational :return: bool

printc (*output, color, flush, stderr*)
info: Will print to the console in color :param output: str :param color: COLOR :param flush: bool :param stderr: std :return:

reset_colors_to_console_defaults ()
info: will reset colors to console defaults :return: None

set_color_16 (*spot, color*)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_16_pal_256 (*spot, color*)
info: sets a color value :param spot: int :param color: int :return: None

set_color_2 (*spot, color*)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_256 (*spot, color*)
info: sets a color value :param spot: int :param color: RGBA :return: None

set_mode (*mode*)
info: will set the color mode :param mode: int: key to color mode :return:

set_tam_color_defaults ()
info: will set console colors to tam defaults :return: None

show_console_cursor (*show*)
info: Will show or hide console cursor :param show: int :return: None

start ()
info: operations for IO to start :return: None

utilities_driver_operational ()
info: checks if the utilities driver is operational :return: bool

tamcolors.tam_io.tam_buffer module

class tamcolors.tam_io.tam_buffer.**TAMBuffer** (*width, height, char, foreground_color, background_color*)
Bases: *tamcolors.utils.object_packer.FastHandObjectPacker*

clear ()
info: clears TAMBuffer :return:

copy ()
info: copy's TAMBuffer :return: TAMBuffer

draw_onto (*tam_buffer, start_x=0, start_y=0, buffer_start_x=0, buffer_start_y=0, buffer_size_x=-1, buffer_size_y=-1, override_alpha=False*)
info: will draw tam_buffer or part of a TAMBuffer onto another TAMBuffer :param tam_buffer: TAM-Buffer :param start_x: int :param start_y: int :param buffer_start_x: int :param buffer_start_y: int :param buffer_size_x: int: 0 - inf :param buffer_size_y: int: 0 - inf :param override_alpha: bool :return:

classmethod from_bytes (*object_byte_array*)
info: from bytes to object :param object_byte_array: bytearray :return: object

get_cross_rect (*tam_buffer, start_x=0, start_y=0, buffer_start_x=0, buffer_start_y=0, buffer_size_x=-1, buffer_size_y=-1*)
info: will draw tam_buffer or part of a TAMBuffer onto another TAMBuffer :param tam_buffer: TAM-Buffer :param start_x: int :param start_y: int :param buffer_start_x: int :param buffer_start_y: int :param buffer_size_x: int: 0 - inf :param buffer_size_y: int: 0 - inf :return:

get_defaults ()
info: gets defaults :return: (str, int, int)

get_dimensions ()
info: gets buffer dimensions :return: (int, int)

get_from_raw_spot (*spot*)
info: gets spot info :param spot: x: int: 0 - (len(tam_buffer) - 1) :return: (int, int, int) or None

get_raw_buffers ()
info: gets raw buffers :return: (list, list, list)

get_raw_spot (*x, y*)
info: return -1 if not a spot :param x: int :param y: int :return: int

get_spot (*x, y*)
info: gets spot info :param x: int :param y: int :return: (int, int, int) or None

replace_alpha_chars (*alpha_replacement=None*)

Parameters **alpha_replacement** – None or str

Returns

set_defaults_and_clear (*char, foreground_color, background_color*)
info: clears buffer and resets defaults :param char: str: len of 1 :param foreground_color: int: 0 - inf :param background_color: int: 0 - inf :return:

set_dimensions_and_clear (*width, height*)
info: clears and resizes TAMBuffer :param width: int: 0 - inf :param height: int: 0 - inf :return:

set_spot (*x, y, char, foreground_color, background_color, override_alpha=False*)
info: sets a single spot on the buffer :param x: int :param y: int :param char: str: len of 1 :param foreground_color: int :param background_color: int :param override_alpha: bool :return:

to_bytes ()
info: object to bytes :return: bytes

tamcolors.tam_io.tam_colors module

class tamcolors.tam_io.tam_colors.**Color** (*mode_16, mode_256, mode_rgb, mode_16_pal_256=None, mode_2=None*)
Bases: *tamcolors.utils.immutable_cache.ImmutableCache, tamcolors.utils.object_packer.FastHandObjectPacker*

classmethod from_bytes (*object_byte_array*)
info: from bytes to object :param object_byte_array: bytearray :return: object

has_alpha
info: Checks if color has any alpha :return: bool

mode_16
info: Gets mode 16 :return: int

mode_16_pal_256
info: Gets mode 16 pal 256 :return: int

mode_2
info: Gets mode 2 :return: int

mode_256
info: Gets mode 256 :return: int

```
mode_rgb
    info: Gets mode rgb :return: RGBA

place_color_over (old_color, override_alpha)
    info: Will calculate what the new color will be :param old_color: Color :param override_alpha: bool
    :return: color

to_bytes ()
    info: object to bytes :return: bytes

transparent_value

class tamcolors.tam_io.tam_colors.RGBA (r, g, b, a=255, is_default=False)
    Bases: tamcolors.utils.immutable_cache.ImmutableCache, tamcolors.utils.
            object_packer.FastHandObjectPacker

    a
        info: Will get the a value :return: int

    b
        info: Will get the b value :return: int

    classmethod from_bytes (object_byte_array)
        info: from bytes to object :param object_byte_array: bytearray :return: object

    g
        info: Will get the g value :return: int

    is_default
        info: See if color is default :return: bool

    r
        info: Will get the r value :return: int

    to_bytes ()
        info: object to bytes :return: bytes
```

tamcolors.tam_io.tam_drivers module

```
class tamcolors.tam_io.tam_drivers.ColorChangerDriver (identifier, mode_2=True,
                                                         mode_16_pal_256=True,
                                                         mode_16=True,
                                                         mode_256=True,
                                                         mode_rgb=True,
                                                         key_driver_operational=True,
                                                         color_driver_operational=True,
                                                         color_changer_driver_operational=True,
                                                         utili-
                                                         ties_driver_operational=True)
    Bases: tamcolors.tam_io.tam_drivers.TAMDriver, abc.ABC

    get_color_16 (spot)
        info: Will get color from color palette 16 :param spot: int :return: RGBA

    get_color_16_pal_256 (spot)
        info: Will get color from color palette 16 :param spot: int :return: int

    get_color_2 (spot)
        info: Will get color from color palette 2 :param spot: int :return: RGBA
```

```

get_color_256 (spot)
    info: Will get color from color palette 256 :param spot: int :return: RGBA

set_color_16 (spot, color)
    info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_16_pal_256 (spot, color)
    info: sets a color value :param spot: int :param color: int :return: None

set_color_2 (spot, color)
    info: sets a color value :param spot: int :param color: RGBA :return: None

set_color_256 (spot, color)
    info: sets a color value :param spot: int :param color: RGBA :return: None

set_mode (mode)
    info: will set the color mode :param mode: int: key to color mode :return:

class tamcolors.tam_io.tam_drivers.ColorDriver (identifier, mode_2=True,
                                                mode_16_pal_256=True,
                                                mode_16=True, mode_256=True,
                                                mode_rgb=True,
                                                key_driver_operational=True,
                                                color_driver_operational=True,
                                                color_changer_driver_operational=True,
                                                utilities_driver_operational=True)

Bases: tamcolors.tam_io.tam_drivers.TAMDriver, abc.ABC

draw (tam_buffer)
    info: Will draw TAMBuffer to console :param tam_buffer: TAMBuffer :return: None

get_printc_mode ()
    Gets the modes used by printc and inputc :return: str

inputc (output, color)
    info: Will get input from the console in color :param output: str :param color: COLOR :return: str

printc (output, color, flush, stderr)
    info: Will print to the console in color :param output: str :param color: COLOR :param flush: bool :param
    stderr: std :return:

set_mode (mode)
    info: will set the color mode :param mode: int: key to color mode :return:

class tamcolors.tam_io.tam_drivers.FullColorDriver (identifier, mode_2=True,
                                                mode_16_pal_256=True,
                                                mode_16=True, mode_256=True,
                                                mode_rgb=True,
                                                key_driver_operational=True,
                                                color_driver_operational=True,
                                                color_changer_driver_operational=True,
                                                utili-
                                                ties_driver_operational=True)

Bases: tamcolors.tam_io.tam_drivers.ColorDriver, tamcolors.tam_io.
tam_drivers.ColorChangerDriver, abc.ABC

```

```
class tamcolors.tam_io.tam_drivers.KeyDriver (identifier, mode_2=True,
                                             mode_16_pal_256=True, mode_16=True,
                                             mode_256=True, mode_rgb=True,
                                             key_driver_operational=True,
                                             color_driver_operational=True,
                                             color_changer_driver_operational=True,
                                             utilities_driver_operational=True)

Bases: tamcolors.tam_io.tam_drivers.TAMDriver, abc.ABC

enable_console_keys (enable)
    info: will enable console keys :param enable: boool :return: None

get_key ()
    info: Gets an input from the terminal :return: tuple or false

static get_key_dict ()
    info: Gets a dict of all the keys :return: {str: (str, str), ... }

class tamcolors.tam_io.tam_drivers.TAMDriver (identifier, mode_2=True,
                                             mode_16_pal_256=True, mode_16=True,
                                             mode_256=True, mode_rgb=True,
                                             key_driver_operational=True,
                                             color_driver_operational=True,
                                             color_changer_driver_operational=True,
                                             utilities_driver_operational=True)

Bases: tamcolors.tam_io.io_tam.IO, abc.ABC

classmethod able_to_execute ()
    info: checks that io is stable in current environment :return: bool

done ()
    info: operations for IO to stop :return: None

start ()
    info: operations for IO to start :return: None

class tamcolors.tam_io.tam_drivers.UtilitiesDriver (identifier, mode_2=True,
                                             mode_16_pal_256=True,
                                             mode_16=True, mode_256=True,
                                             mode_rgb=True,
                                             key_driver_operational=True,
                                             color_driver_operational=True,
                                             color_changer_driver_operational=True,
                                             utili-
                                             ties_driver_operational=True)

Bases: tamcolors.tam_io.tam_drivers.TAMDriver, abc.ABC

clear ()
    info: Will clear the console :return: None

get_dimensions ()
    info: Gets the dimensions of console :return: (int, int): (row, column)

show_console_cursor (show)
    info: Will show or hide console cursor :param show: int :return: None
```


tamcolors.tam_io.tam_identifier module

```

class tamcolors.tam_io.tam_identifier.TAMIdentifier(name, *drivers)
    Bases: object

    get_all_drivers()
        info: Get a list of all the tam drivers being used :return: list

    get_info_dict()
        info: Get raw info about this identifier :return: dict

    get_io()
        info: Gets the IO that the selected drivers made :return: IO

    get_name()
        info: Get the general name of the drivers :return: str

    get_system()
        info: Get the system :return: str

    classmethod identify()
        info: Will find the best drivers for current environment :return: TAMIdentifier

    stable()
        info: Sees if this IO is able to execute :return: bool

```

tamcolors.tam_io.tam_keys module**tamcolors.tam_io.tcp_io module**

```

tamcolors.tam_io.tcp_io.get_tcp_io(receiver)
    info: will return TCPObjectConnector connected to an io object :param receiver: TCPReceiver :return: TCPOb-
    jectConnector

```

tamcolors.tam_io.uni_drivers module

```

class tamcolors.tam_io.uni_drivers.UNIKeyDriver(*args, **kwargs)
    Bases: tamcolors.tam_io.tam_drivers.KeyDriver, tamcolors.tam_io.uni_drivers.
    UNISharedData, abc.ABC

    enable_console_keys(enable)
        info: will enable console keys :param enable: boool :return: None

    get_key()
        info: Gets an input from the terminal :return: tuple or false

    static get_key_dict()
        info: Gets a dict of all the keys :return: {str: (str, str), ... }

class tamcolors.tam_io.uni_drivers.UNISharedData(identifier,
                                                    mode_2=True,
                                                    mode_16_pal_256=True,
                                                    mode_16=True, mode_256=True,
                                                    mode_rgb=True,
                                                    key_driver_operational=True,
                                                    color_driver_operational=True,
                                                    color_changer_driver_operational=True,
                                                    utilities_driver_operational=True)

```

Bases: *tamcolors.tam_io.tam_drivers.TAMDriver*, *abc.ABC*

classmethod **able_to_execute**()

info: checks that io is stable in current environment :return: bool

```
class tamcolors.tam_io.uni_drivers.UNIUtilitiesDriver(identifier,      mode_2=True,  
                                                    mode_16_pal_256=True,  
                                                    mode_16=True,  
                                                    mode_256=True,  
                                                    mode_rgb=True,  
                                                    key_driver_operational=True,  
                                                    color_driver_operational=True,  
                                                    color_changer_driver_operational=True,  
                                                    utili-  
                                                    ties_driver_operational=True)
```

Bases: *tamcolors.tam_io.tam_drivers.UtilitiesDriver*, *tamcolors.tam_io.uni_drivers.UNISharedData*, *abc.ABC*

clear()

info: Will clear the console :return: None

done()

info: operations for IO to stop :return: None

get_dimensions()

info: Gets the dimensions of console :return: (int, int): (row, column)

show_console_cursor(*show*)

info: Will show or hide console cursor :param show: int :return: None

tamcolors.tam_io.win_drivers module

```
class tamcolors.tam_io.win_drivers.WINFullColorDriver(*args, **kwargs)
```

Bases: *tamcolors.tam_io.tam_drivers.FullColorDriver*, *tamcolors.tam_io.win_drivers.WinSharedData*, *abc.ABC*

done()

info: operations for IO to stop :return: None

draw(*tam_buffer*)

info: will draw tam buffer to terminal :param tam_buffer: TAMBuffer :return:

get_printc_mode()

Gets the modes used by printc and inputc :return: str

inputc(*output, color*)

info: will get user input with color :param output: str :param color: tuple: (int, int) :return: str

printc(*output, color, flush, stderr*)

info: will print out user output with color :param output: str :param color: tuple: (int, int) :param flush: boolean :param stderr: boolean :return: None

set_mode(*mode*)

info: will set the color mode :param mode: int: key to color mode :return:

```
class tamcolors.tam_io.win_drivers.WINKeyDriver(*args, **kwargs)
```

Bases: *tamcolors.tam_io.tam_drivers.KeyDriver*, *tamcolors.tam_io.win_drivers.WinSharedData*, *abc.ABC*

```

get_key()
    info: will get single key input or return False :return: str or False

static get_key_dict()
    info: makes a dict mapping key codes to key :return: dict

class tamcolors.tam_io.win_drivers.WINUtilitiesDriver(*args, **kwargs)
    Bases: tamcolors.tam_io.tam_drivers.UtilitiesDriver, tamcolors.tam_io.win_drivers.WinSharedData, abc.ABC

    clear()
        info: will clear the screen :return:

    done()
        info: operations for IO to stop :return: None

    get_dimensions()
        info: will get teh terminal dimensions :return: (int, int)

    show_console_cursor(show)
        info: Will show or hide console cursor :param show: bool :return: None

    start()
        info: operations for IO to start :return: None

class tamcolors.tam_io.win_drivers.WinSharedData(identifier, mode_2=True,
                                                mode_16_pal_256=True,
                                                mode_16=True, mode_256=True,
                                                mode_rgb=True,
                                                key_driver_operational=True,
                                                color_driver_operational=True,
                                                color_changer_driver_operational=True,
                                                utilities_driver_operational=True)
    Bases: tamcolors.tam_io.tam_drivers.TAMDriver, abc.ABC

    classmethod able_to_execute()
        info: checks that io is stable in current environment :return: bool

```

Module contents

tamcolors.tam_tools package

Submodules

tamcolors.tam_tools.tam_color_palette module

```

class tamcolors.tam_tools.tam_color_palette.TAMColorPalette(color_range=range(0,
                                                            16),
                                                            color_rules=None)

    Bases: object

    get_color(key)
        info: gets a color from the color palette :param key: object :return:

    get_rule(key)
        info: gets a instance of TAMColorPaletteRule or None :param key: object :return: instance of TAMColor-
        PaletteRule or None

```

```
key_present (key)
    info: checks if key is in color palette :param key: object :return: bool

set_color (key, color)
    info sets a color to the color palette :param key: object :param color: Color :return:

set_rule (key, rule)
    info: sets a color rule :param key: object :param rule: instance of TAMColorPaletteRule :return:

update ()
    info: updates color_palette rules :return:

exception tamcolors.tam_tools.tam_color_palette.TAMColorPaletteError
    Bases: Exception

class tamcolors.tam_tools.tam_color_palette.TAMColorPaletteRule
    Bases: object

    update (color_palette, key)

class tamcolors.tam_tools.tam_color_palette.TAMCycleColor (colors, clock=1)
    Bases: tamcolors.tam_tools.tam_color_palette.TAMColorPaletteRule

    get_clock ()
        info: gets the clock rate :return: int

    set_clock (clock)
        info: sets the clock :param clock: int :return: int: 1 - inf

    set_colors (colors)
        info: sets all the colors :param colors: tuple or int: [Color, Color, Color, ...] :return:

    update (color_palette, key)
        info: will update the color_palette color :param color_palette: TAMColorPalette :param key: object :return:

class tamcolors.tam_tools.tam_color_palette.TAMDefaultColor (color)
    Bases: tamcolors.tam_tools.tam_color_palette.TAMColorPaletteRule

    get_color ()
        info: gets the default color :return: Color

    set_color (color)
        info: sets the default color :param color: Color :return:

    update (color_palette, key)
        info: will update the color_palette color :param color_palette: TAMColorPalette :param key: object :return:
```

tamcolors.tam_tools.tam_fade module

```
tamcolors.tam_tools.tam_fade.tam_fade_in (buffer, char, foreground_color, background_color, rand=(True, False, False, False, False), reverse=False)

info: makes a fade in or fade out via TAMFilm :param buffer: TAMBuffer :param char: single block char :param foreground_color: Color :param background_color: Color :param rand: list: [True, bool, bool, bool, ...] :param reverse: bool :return: TAMFilm
```

tamcolors.tam_tools.tam_film module

```

class tamcolors.tam_tools.tam_film.TAMFilm(tam_buffers=None, circular=False)
    Bases: object

    append(tam_buffer)
        info :param tam_buffer: :return:

    done()
        info: returns true if film is done :return: bool

    get(spot)
        info will get a TAMBuffer :param spot: int: 0 - len(self.__buffer_list) :return: TAMBuffer

    get_circular()
        info: gets the circular value :return: bool

    peak()
        info: gets the current frame :return: TAMBuffer or None

    pop()
        info: will pop the last TAMBuffer :return: TAMBuffer or None

    set(spot, tam_buffer)
        info: sets a frame :param spot: int: 0 - len(self.__buffer_list) :param tam_buffer: TAMBuffer :return:

    set_circular(circular)
        info: can set circular value :param circular: True :return:

    slide()
        info will get the next TAMBuffer :return: TAMBuffer or None

exception tamcolors.tam_tools.tam_film.TAMFilmError
    Bases: Exception

```

tamcolors.tam_tools.tam_icon module

```

tamcolors.tam_tools.tam_icon.get_icon()

```

tamcolors.tam_tools.tam_key_manager module

```

class tamcolors.tam_tools.tam_key_manager.TAMKeyManager (all_keys=({'t', 'WHITES-
PACE'), ('n', 'WHITES-
PACE'), (' ', 'WHITES-
PACE'), ('!', 'NORMAL'),
('"' , 'NORMAL'), ('#',
'NORMAL'), ('$' , 'NOR-
MAL'), ('%' , 'NOR-
MAL'), ('&' , 'NORMAL'),
('"' , 'NORMAL'), ('(' ,
'NORMAL'), (')' , 'NOR-
MAL'), ('*' , 'NORMAL'),
('+', 'NORMAL'), (',
', 'NORMAL'), ('-' ,
'NORMAL'), (',' , 'NOR-
MAL'), ('/' , 'NORMAL'),
('0' , 'NORMAL'), ('1' ,
'NORMAL'), ('2' , 'NOR-
MAL'), ('3' , 'NORMAL'),
('4' , 'NORMAL'), ('5' ,
'NORMAL'), ('6' , 'NOR-
MAL'), ('7' , 'NORMAL'),
('8' , 'NORMAL'), ('9' ,
'NORMAL'), (':' , 'NOR-
MAL'), ( ';' , 'NORMAL'),
('<' , 'NORMAL'), ('=' ,
'NORMAL'), ('>' , 'NOR-
MAL'), ('?' , 'NORMAL'),
('@' , 'NORMAL'), ('A' ,
'NORMAL'), ('B' , 'NOR-
MAL'), ('BACKSPACE' ,
'SPECIAL'), ('C' , 'NOR-
MAL'), ('D' , 'NORMAL'),
('DELETE' , 'SPECIAL'),
('DOWN' , 'SPECIAL'),
('E' , 'NORMAL'), ('ES-
CAPE' , 'SPECIAL'),
('F' , 'NORMAL'), ('F1' ,
'SPECIAL'), ('F12' , 'SPE-
CIAL'), ('F12_SHIFT' ,
'SPECIAL'), ('F1_SHIFT' ,
'SPECIAL'), ('F2' , 'SPE-
CIAL'), ('F2_SHIFT' ,
'SPECIAL'), ('F3' , 'SPE-
CIAL'), ('F3_SHIFT' ,
'SPECIAL'), ('F4' , 'SPE-
CIAL'), ('F4_SHIFT' ,
'SPECIAL'), ('F5' , 'SPE-
CIAL'), ('F5_SHIFT' ,
'SPECIAL'), ('F6' , 'SPE-
CIAL'), ('F6_SHIFT' ,
'SPECIAL'), ('F7' , 'SPE-
CIAL'), ('F7_SHIFT' ,
'SPECIAL'), ('F8' , 'SPE-
CIAL'), ('F8_SHIFT' ,
'SPECIAL'), ('F9' , 'SPE-
CIAL'), ('F9_SHIFT' ,
'SPECIAL'), ('G' , 'NOR-
MAL'), ('H' , 'NORMAL')

```

```

get_key_state (key)
    info: will get a state of a key :param key: str :return: int

get_raw_user_input ()
    info: will get the raw user input :return: list or tuple: [(str, str), ...]

get_user_input ()
    info: will the next key the user enters :return: tuple, list or None: (str, str)

get_user_input_generator ()
    info: yields a key at a time :return: list or tuple: (str, str)

silent_key_state (key)
    info: will get a key state and make it 0 :param key: str :return: int

update (keys)
    info: will update key manager with next set of keys :param keys: tuple or list: [(str, str), ...] :return:

```

tamcolors.tam_tools.tam_list_buffer module

```

tamcolors.tam_tools.tam_list_buffer.tam_list_buffer (chars, foreground_colors, back-
                                                    ground_colors)
    info: makes a list into a TAMBuffer :param chars: list, tuple :param foreground_colors: list, tuple, int :param
    background_colors: list, tuple, int :return: TAMBuffer

```

tamcolors.tam_tools.tam_menu module

```

class tamcolors.tam_tools.tam_menu.TAMButtonRule
    Bases: object

    draw (buffer)

    get_action ()

    get_position ()

    off ()

    on ()

    run_action ()

    set_action (func)

    set_position (x, y)

    update ()

class tamcolors.tam_tools.tam_menu.TAMMenu (buttons, call_key, goto_map, on=0)
    Bases: object

    draw (buffer)
        info: draw all the buttons of the menu :param buffer: TAMBuffer :return:

    get_buttons ()
        info: gets the buttons :return: list

    get_call_key ()
        info: gets the call key :return: str

    get_goto_map ()
        info: gets the goto map :return: dict

```

```
get_on ()
    info: gets the on value :return: int

static simple_menu_builder (buttons, call_on, up_keys=('UP', ), down_keys=('DOWN', ),
                             on=0)
    info: a simple way of making a TAMMenu :param buttons: TAMButtonRule :param call_on: str :param
    up_keys: list or tuple: (str, ...) :param down_keys: list or tuple: (str, ...) :param on: int :return:
    TAMMenu

update (keys)
    info: update the menu :param keys: list or tuple: [(str, str), (str, str, ...), ...] :return:

class tamcolors.tam_tools.tam_menu.TAMTextBoxButton (text, x, y, width, height,
                                                    char, foreground_color,
                                                    background_color, action_func, on_foreground_color,
                                                    on_background_color,
                                                    on_char='/', clock=-1,
                                                    center_vertical=True, center_horizontal=False,
                                                    vertical_space=1, vertical_start=1,
                                                    char_background='')

Bases: tamcolors.tam_tools.tam_menu.TAMButtonRule

draw (buffer)
    info: draws button onto buffer :param buffer: :return:

get_action ()
    info: gets action function :return: fuction

get_position ()
    info: gets the position :return:

off ()
    info: puts the button onto the off state :return:

on ()
    info: puts the button onto the on state :return:

run_action ()
    " info: calls the action function :return:

set_action (func)
    info: sets the actions :param func: function :return:

set_position (x, y)
    info: sets the position :param x: int :param y: int :return:

update ()
    info: updates the button :return:

class tamcolors.tam_tools.tam_menu.TAMTextButton (text, x, y, foreground_color,
                                                    background_color,
                                                    action_func, on_foreground_color,
                                                    on_background_color, on_chars='*
                                                    ')

Bases: tamcolors.tam_tools.tam_menu.TAMButtonRule

draw (buffer)
    info: draws button onto buffer :param buffer: :return:
```



```

get_action()
    info: gets action function :return: fuction

get_position()
    info: gets the position :return:

off()
    info: puts the button onto the off state :return:

on()
    info: puts the button onto the on state :return:

run_action()
    " info: calls the action function :return:

set_action(func)
    info: sets the actions :param func: function :return:

set_position(x, y)
    info: sets the position :param x: int :param y: int :return:

update()
    info: updates the button :return:

```

tamcolors.tam_tools.tam_placing module

```

tamcolors.tam_tools.tam_placing.bottom_left(x, y, width, height)
    info: gets the bottom left x, y :param x: int :param y: int :param width: int: 0 - inf :param height: int: 0 - inf
    :return: (int, int)

tamcolors.tam_tools.tam_placing.bottom_right(x, y, width, height)
    info: gets the bottom right x, y :param x: int :param y: int :param width: int: 0 - inf :param height: int: 0 - inf
    :return: (int, int)

tamcolors.tam_tools.tam_placing.center(x, y, width, height)
    info: gets the center x, y :param x: int :param y: int :param width: int: 0 - inf :param height: int: 0 - inf :return:
    (int, int)

tamcolors.tam_tools.tam_placing.top_left(x, y, width, height)
    info: gets the top left x, y :param x: int :param y: int :param width: int: 0 - inf :param height: int: 0 - inf :return:
    (int, int)

tamcolors.tam_tools.tam_placing.top_right(x, y, width, height)
    info: gets the top right x, y :param x: int :param y: int :param width: int: 0 - inf :param height: int: 0 - inf
    :return: (int, int)

```

tamcolors.tam_tools.tam_print module

```

tamcolors.tam_tools.tam_print.tam_print(tam_buffer, x, y, text, foreground_color,
                                         background_color, error_bad_char=False,
                                         bad_char="")
    info: tam_print will draw string on a tam_buffer :param tam_buffer: TAMBuffer :param x: int :param y: int
    :param text: object with __str__ :param foreground_color: int: -1 - inf: use current foreground_color :param
    background_color: int: -1 - inf: use current background_color :param error_bad_char: bool :param bad_char:
    str :return:

```

tamcolors.tam_tools.tam_str module

exception tamcolors.tam_tools.tam_str.**TAMStrError**

Bases: Exception

tamcolors.tam_tools.tam_str.**make_tam_str**(*text*, *end_line*='\n', *bad_char*=None)

info: formats str into a tam str :param text: str :param end_line: str :param bad_char: str :return: str

tamcolors.tam_tools.tam_text_box module

class tamcolors.tam_tools.tam_text_box.**TAMTextBox**(*text*, *width*, *height*, *char*, *foreground_color*, *background_color*, *clock*=-1, *center_vertical*=True, *center_horizontal*=False, *vertical_space*=1, *vertical_start*=1, *char_background*='')
Bases: object

done()

info: True if text box has placed all chars onto to its self :return: bool

draw(*tam_buffer*, *start_x*=0, *start_y*=0)

info: draws the text box on to another buffer :param tam_buffer: TAMBuffer :param start_x: int :param start_y: int :return:

get_char()

info: gets char :return: str

get_colors()

info: gets textbox color :return: (int, int)

get_dimensions()

info: gets buffer dimensions :return: (int, int)

get_text()

info: gets text :return: str

set_char(*char*)

info: sets char :return:

set_colors(*foreground_color*, *background_color*)

info: sets textbox colors :param foreground_color: 0 - inf :param background_color: 0 - inf :return:

update()

info updates the text box :return:

Module contents

tamcolors.tests package

Subpackages

tamcolors.tests.tam_basic_tests package

Submodules

tamcolors.tests.tam_basic_tests.basic_tests module

```

class tamcolors.tests.tam_basic_tests.basic_tests.ClearTests (methodName='runTest')
    Bases: unittest.case.TestCase

    static test_clear()

class tamcolors.tests.tam_basic_tests.basic_tests.ColorNameToCodeTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_color_name_to_color_1()

    test_color_name_to_color_2()

class tamcolors.tests.tam_basic_tests.basic_tests.GetColorCodeTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test__get_color_code_1()

    test__get_color_code_2()

    test__get_color_code_3()

    test__get_color_code_4()

class tamcolors.tests.tam_basic_tests.basic_tests.IOTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_IO()

class tamcolors.tests.tam_basic_tests.basic_tests.InputCTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_inputc()

class tamcolors.tests.tam_basic_tests.basic_tests.PrintCTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_flush_and_stderr()

    test_not_same_color()

    static test_same_color()

```

Module contents**tamcolors.tests.tam_io_tests package****Submodules****tamcolors.tests.tam_io_tests.any_drivers_tests module**

```

class tamcolors.tests.tam_io_tests.any_drivers_tests.AnyIOTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_get_dimensions()

    test_get_io()

    test_get_key()

    test_get_modes()

```

```
    static test_reset_colors_to_console_defaults()
    test_set_slash_get_mode()
    static test_set_tam_color_defaults()
tamcolors.tests.tam_io_tests.any_drivers_tests.get_any_io()
```

tamcolors.tests.tam_io_tests.io_tam_tests module

```
class tamcolors.tests.tam_io_tests.io_tam_tests.IOTAMTest (methodName='runTest')
    Bases: unittest.case.TestCase
    test__draw_onto()
    test__draw_onto_2()
    test__draw_onto_3()
    static test__write_to_output_stream()
    static test__write_to_output_stream_2()
class tamcolors.tests.tam_io_tests.io_tam_tests.RAWIOTest (methodName='runTest')
    Bases: unittest.case.TestCase
    test_same_doc_string()
    test_same_methods()
```

tamcolors.tests.tam_io_tests.tam_buffer_tests module

```
class tamcolors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests (methodName='runTest')
    Bases: unittest.case.TestCase
    test_buffer_eq()
    test_buffer_eq_2()
    test_buffer_eq_3()
    test_buffer_eq_4()
    test_buffer_eq_5()
    test_buffer_init()
    test_buffer_len()
    test_buffer_len_2()
    test_buffer_len_3()
    test_buffer_len_4()
    test_buffer_ne()
    test_buffer_ne_2()
    test_buffer_ne_3()
    test_buffer_ne_4()
    test_buffer_str()
```

```
test_buffer_str_2()
test_buffer_str_3()
test_clear()
test_copy()
test_copy_2()
test_draw_onto()
test_draw_onto_2()
test_draw_onto_3()
test_draw_onto_4()
test_draw_onto_5()
test_draw_onto_6()
test_draw_onto_7()
test_draw_onto_8()
test_get_cross_rect()
test_get_cross_rect_2()
test_get_defaults()
test_get_dimensions()
test_get_dimensions_2()
test_get_dimensions_3()
test_get_from_raw_spot()
test_get_raw_buffers()
test_get_raw_buffers_2()
test_get_raw_spot()
test_get_spot()
test_replace_alpha_chars_1()
test_replace_alpha_chars_2()
test_replace_alpha_chars_3()
test_set_defaults_and_clear()
test_set_defaults_and_clear_2()
test_set_dimensions_and_clear()
test_set_dimensions_and_clear_2()
test_set_spot()
test_set_spot_2()
test_set_spot_3()
test_set_spot_4()
test_set_spot_5()
```

```
test_set_spot_6()
test_set_spot_7()
test_set_spot_8()
test_set_spot_9()
```

tamcolors.tests.tam_io_tests.tam_colors_tests module

```
class tamcolors.tests.tam_io_tests.tam_colors_tests.ColorTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_color_eq()
    test_color_eq_2()
    test_color_eq_3()
    test_color_init()
    test_color_init_2()
    test_color_ne()
    test_color_ne_2()
    test_get_property()
    test_get_property_2()

class tamcolors.tests.tam_io_tests.tam_colors_tests.RGBATests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_get_property()
    test_get_property_2()
    test_rgba_eq()
    test_rgba_eq_2()
    test_rgba_eq_3()
    test_rgba_init()
    test_rgba_init_2()
    test_rgba_ne()
    test_rgba_ne_2()

class tamcolors.tests.tam_io_tests.tam_colors_tests.TAMColorTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_colors()
```

tamcolors.tests.tam_io_tests.tam_keys_tests module

```
class tamcolors.tests.tam_io_tests.tam_keys_tests.TAMKeyTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_keys()
```

tamcolors.tests.tam_io_tests.tam_standard_tests module

```
class tamcolors.tests.tam_io_tests.tam_standard_tests.StandardTest (methodName='runTest')
    Bases: unittest.case.TestCase

    test_same_keys()
```

tamcolors.tests.tam_io_tests.uni_drivers_tests module

```
class tamcolors.tests.tam_io_tests.uni_drivers_tests.UniIOTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_able_to_execute()
    test_clear()
    static test_done()
    test_get_dimensions()
    test_get_key()
    test_get_key_2()
    test_get_key_3()
    test_get_key_4()
    test_get_key_5()
    test_get_key_dict()
    test_get_modes()
    static test_hide_console_cursor()
    test_set_slash_get_mode()
    static test_show_console_cursor()
    static test_start()

tamcolors.tests.tam_io_tests.uni_drivers_tests.get_uni_io()
```

tamcolors.tests.tam_io_tests.win_drivers_tests module

```
class tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test__console_color_count()
    test__draw_16()
    static test__draw_2()
    test__get_buffer_dimension()
    static test__print()
    static test__print_2()
    static test__print_3()
    test__processes_special_color_1()
```

```
test__processes_special_color_2()
test__processes_special_color_3()
test__processes_special_color_4()
test__processes_special_color_5()
test__processes_special_color_6()
test__spot_swap()
test_able_to_execute()
test_done()
test_get_dimensions()
test_get_key()
test_get_key_2()
test_get_key_3()
test_get_key_4()
test_get_key_5()
test_get_key_dict()
test_get_modes()
test_reset_colors_to_console_defaults()
test_set_slash_get_mode()
test_set_tam_color_defaults()
test_start()

class tamcolors.tests.tam_io_tests.win_drivers_tests.WinGlobalsTests (methodName='runTest')
    Bases: unittest.case.TestCase
        test_win_stable()

tamcolors.tests.tam_io_tests.win_drivers_tests.get_win_io()
```

Module contents

tamcolors.tests.tam_tests package

Submodules

tamcolors.tests.tam_tests.tam_loop_tests module

```
class tamcolors.tests.tam_tests.tam_loop_tests.TAMFrameTests (methodName='runTest')
    Bases: unittest.case.TestCase
        test__done()
        test__done_2()
        test_draw()
        test_frame_init()
```



```

test_get_defaults()
test_get_defaults_2()
test_get_fps()
test_get_fps_2()
test_get_height_min_and_max()
test_get_height_min_and_max_2()
test_get_width_min_and_max()
test_get_width_min_and_max_2()
test_make_buffer_ready()
test_make_buffer_ready_2()
test_make_buffer_ready_3()
test_update()
class tamcolors.tests.tam_tests.tam_loop_tests.TAMLoopTests (methodName='runTest')
    Bases: unittest.case.TestCase
    test_loop_init()
    test_preferred_mode()
    test_preferred_mode_2()
    test_preferred_mode_3()
    test_reset_colors_to_console_defaults()
    test_run()
    test_set_tam_color_defaults()
    test_stack()
    test_step()

```

Module contents

tamcolors.tests.tam_tools_tests package

Submodules

tamcolors.tests.tam_tools_tests.tam_color_palette_tests module

```

class tamcolors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests (methodName='runTest')
    Bases: unittest.case.TestCase
    test_get_color()
    test_get_color_2()
    test_get_color_3()
    test_get_rule()
    test_get_rule_2()

```

```
test_getitem()
test_getitem_2()
test_getitem_3()
test_init_color_palette()
test_init_color_palette_2()
test_key_present()
test_key_present_2()
test_set_color()
test_set_color_2()
test_set_color_3()
test_set_rule()
test_set_rule_2()
test_setitem()
test_setitem_2()
test_setitem_3()
test_str()
test_str_2()
test_update()

class tamcolors.tests.tam_tools_tests.tam_color_palette_tests.TAMCycleColorTests (methodName=
    Bases: unittest.case.TestCase
    test_get_clock()
    test_init_cycle_color()
    test_set_clock()
    test_set_colors()
    test_update()
    test_update_2()

class tamcolors.tests.tam_tools_tests.tam_color_palette_tests.TAMDefaultColorTests (methodName=
    Bases: unittest.case.TestCase
    test_get_color()
    test_init_default_color()
    test_set_color()
    test_update()
```

tamcolors.tests.tam_tools_tests.tam_fade_tests module

```
class tamcolors.tests.tam_tools_tests.tam_fade_tests.TAMFilmFadeInTests (methodName='runTest')
    Bases: unittest.case.TestCase
    test_tam_fade_in()
```

```
test_tam_fade_in_2()  
test_tam_fade_in_3()  
test_tam_fade_in_4()
```

tamcolors.tests.tam_tools_tests.tam_film_tests module

```
class tamcolors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_append()  
    test_done()  
    test_done_2()  
    test_done_3()  
    test_get()  
    test_get_2()  
    test_get_3()  
    test_get_circular()  
    test_get_circular_2()  
    test_getitem()  
    test_getitem_2()  
    test_getitem_3()  
    test_init_film()  
    test_len()  
    test_len_2()  
    test_next()  
    test_next_2()  
    test_peak()  
    test_peak_2()  
    test_peak_3()  
    test_pop()  
    test_set()  
    test_set_2()  
    test_set_3()  
    test_set_circular()  
    test_setitem()  
    test_setitem_2()  
    test_setitem_3()  
    test_slide()
```

```
test_slide_2()
```

tamcolors.tests.tam_tools_tests.tam_key_manager_tests module

```
class tamcolors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests (methodName='runTest')
    Bases: unittest.case.TestCase
    get_raw_user_input_2()
    get_raw_user_input_3()
    test_get_key_state()
    test_get_key_state_2()
    test_get_raw_user_input()
    test_get_user_input()
    test_get_user_input_2()
    test_get_user_input_3()
    test_get_user_input_generator()
    test_get_user_input_generator_2()
    test_get_user_input_generator_3()
    test_init_key_manger()
    test_iter()
    test_silent_key_state()
    test_silent_key_state_2()
    test_silent_key_state_3()
    test_update()
    test_update_2()
```

tamcolors.tests.tam_tools_tests.tam_list_buffer_tests module

```
class tamcolors.tests.tam_tools_tests.tam_list_buffer_tests.TAMListBufferTests (methodName='runTest')
    Bases: unittest.case.TestCase
    test_tam_list_buffer()
    test_tam_list_buffer_2()
    test_tam_list_buffer_3()
    test_tam_list_buffer_4()
    test_tam_list_buffer_5()
```

tamcolors.tests.tam_tools_tests.tam_menu_tests module

```

class tamcolors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_draw()
    test_get_buttons()
    test_get_buttons_2()
    test_get_buttons_3()
    test_get_call_key()
    test_get_call_key_2()
    test_get_goto_map()
    test_get_on()
    test_get_on_2()
    test_simple_menu_builder()
    test_simple_menu_builder_2()
    test_simple_menu_builder_3()
    test_simple_menu_builder_4()
    test_simple_menu_builder_5()
    test_tam_menu()
    test_tam_menu_2()
    test_update()
    test_update_2()
    test_update_3()

class tamcolors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests (methodName='runTest')
    Bases: unittest.case.TestCase

    static test_call_TextBoxButton()
    test_draw()
    test_draw_2()
    test_get_action()
    test_get_position()
    test_get_position_2()
    test_init_TextBoxButton()
    test_init_TextBoxButton_2()
    test_off()
    test_off_2()
    test_on()
    test_on_2()

```

```
    static test_run_action()
    test_set_action()
    test_set_position()
    test_set_position_2()
    test_str_TextBoxButton()
    test_str_TextBoxButton_2()
    static test_update()

class tamcolors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests (methodName='runTest')
    Bases: unittest.case.TestCase
    static test_call_TextButton()
    test_draw()
    test_draw_2()
    test_get_action()
    test_get_position()
    test_get_position_2()
    test_init_TextButton()
    test_init_TextButton_2()
    test_off()
    test_off_2()
    test_on()
    test_on_2()
    static test_run_action()
    test_set_action()
    test_set_position()
    test_set_position_2()
    test_str_TextButton()
    test_str_TextButton_2()
    static test_update()
```

tamcolors.tests.tam_tools_tests.tam_placing_tests module

```
class tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests (methodName='runTest')
    Bases: unittest.case.TestCase
    test__get_center()
    test__get_center_2()
    test__get_dimensions_wrapper()
    test__get_dimensions_wrapper_2()
```

```
test__get_dimensions_wrapper_3()  
test__get_other_side()  
test__get_other_side_2()  
test_bottom_left()  
test_bottom_left_2()  
test_bottom_right()  
test_bottom_right_2()  
test_center()  
test_center_2()  
test_top_left()  
test_top_left_2()  
test_top_right()  
test_top_right_2()
```

tamcolors.tests.tam_tools_tests.tam_print_tests module

```
class tamcolors.tests.tam_tools_tests.tam_print_tests.TAMPrintTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_tam_print()  
    test_tam_print_2()  
    test_tam_print_3()  
    test_tam_print_4()  
    test_tam_print_5()
```

tamcolors.tests.tam_tools_tests.tam_str_tests module

```
class tamcolors.tests.tam_tools_tests.tam_str_tests.MakeTAMStrTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_make_tam_str()  
    test_make_tam_str_2()  
    test_make_tam_str_3()  
    test_make_tam_str_4()  
    test_make_tam_str_5()
```

tamcolors.tests.tam_tools_tests.tam_text_box_tests module

```
class tamcolors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_done()
```

```
test_done_2()
test_draw()
test_draw_2()
test_draw_3()
test_get_char()
test_get_char_2()
test_get_colors()
test_get_colors_2()
test_get_dimensions()
test_get_dimensions_2()
test_get_text()
test_get_text_2()
test_set_char()
test_set_char_2()
test_set_colors()
test_set_colors_2()
test_set_colors_3()
test_tam_text_box_init()
test_tam_text_box_str()
test_tam_text_box_str_2()
test_update()
```

Module contents

tamcolors.tests.tests_tests package

Submodules

tamcolors.tests.tests_tests.test_multi_task_helper_tests module

```
class tamcolors.tests.tests_tests.test_multi_task_helper_tests.MultiTaskHelperTests (methodNo
    Bases: tamcolors.tests.test_multi_task_helper.MultiTaskHelper, unittest.case.
    TestCase

    test_assert_no_main_process()
    test_assert_no_main_thread()
    test_assert_process()
    test_assert_thread()
    test_hand_down()
```


Module contents

tamcolors.tests.utils_tests package

Submodules

tamcolors.tests.utils_tests.compress_tests module

```
class tamcolors.tests.utils_tests.compress_tests.CompressTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_1 ()
    test_2 ()
    test_3 ()
    test_4 ()
```

tamcolors.tests.utils_tests.encryption_tests module

```
class tamcolors.tests.utils_tests.encryption_tests.EncryptionTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_build_setting ()
    test_encrypt_decrypt ()
    test_init ()
    test_simple_encrypt_decrypt ()

class tamcolors.tests.utils_tests.encryption_tests.NoEncryptionTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_init ()
```

tamcolors.tests.utils_tests.identifier_tests module

```
class tamcolors.tests.utils_tests.identifier_tests.IdentifierTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_generate_identifier ()
    test_get_identifier_bytes ()
    test_globals ()
```

tamcolors.tests.utils_tests.immutable_cache_tests module

```
class tamcolors.tests.utils_tests.immutable_cache_tests.DummyCache (*args,
                                                                    **kwargs)
    Bases: tamcolors.utils.immutable_cache.ImmutableCache

class tamcolors.tests.utils_tests.immutable_cache_tests.ImmutableCacheTests (methodName='runTest')
    Bases: unittest.case.TestCase
```

```
test_cache()  
test_init()  
test_new()
```

tamcolors.tests.utils_tests.log_tests module

```
class tamcolors.tests.utils_tests.log_tests.LogTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_critical()  
    test_debug()  
    static test_enable_and_disable_logging()
```

tamcolors.tests.utils_tests.object_packer_tests module

```
class tamcolors.tests.utils_tests.object_packer_tests.FastHandObjectPackerTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_dunder_bytes()  
    test_tam_buffer_1()  
    test_tam_buffer_2()  
    test_tam_color_1()  
    test_tam_color_2()  
  
class tamcolors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_save_and_load_data_1()  
    test_save_and_load_data_2()  
    test_save_and_load_data_3()  
    test_save_and_load_data_4()  
    test_save_and_load_int_1()  
    test_save_and_load_int_2()  
    test_save_and_load_int_3()  
    test_save_and_load_int_4()  
  
class tamcolors.tests.utils_tests.object_packer_tests.ObjectPackerJsonTests (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_bool()  
    test_bytearray()  
    test_bytes()  
    test_dict()  
    test_fast_hand_object()  
    test_float()
```

```

test_int()
test_large_data()
test_list()
test_none()
test_set()
test_str()
test_tuple()

```

tamcolors.tests.utils_tests.tcp_tests module

```

class tamcolors.tests.utils_tests.tcp_tests.TCPObjectWrapper (methodName='runTest')
    Bases: tamcolors.tests.test_multi_task_helper.MultiTaskHelper, unittest.case.
    TestCase

    test_object_wrapper()

class tamcolors.tests.utils_tests.tcp_tests.TCPTests (methodName='runTest')
    Bases: tamcolors.tests.test_multi_task_helper.MultiTaskHelper, unittest.case.
    TestCase

    test_ping()
    test_ping_ipv6()

```

tamcolors.tests.utils_tests.timer_tests module

```

class tamcolors.tests.utils_tests.timer_tests.TimerTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_lap()
    test_offset_sleep()
    test_timer()

```

tamcolors.tests.utils_tests.transport_optimizer_tests module

```

class tamcolors.tests.utils_tests.transport_optimizer_tests.TransportOptimizerTests (methodName='runTest')
    Bases: unittest.case.TestCase

    test_large_data()
    test_large_tam_buffer_data()
    test_same_data()
    test_simple_data()

```

Module contents

Submodules

tamcolors.tests.all_tests module

```
tamcolors.tests.all_tests.load_tests (loader, other_tests=None, pattern=None)
tamcolors.tests.all_tests.stability_check (ret_bool=True, run_slow=False)
    info: run all TAM tests :param ret_bool: bool :param run_slow: bool :return: (int, int) or bool: (test_pasted,
    test_ran) or True if all test passed
tamcolors.tests.all_tests.tests_main (run_slow=False)
    info: the main way tamcolors run tests :param run_slow: bool: will run slow tests :return: bool
```

tamcolors.tests.test_multi_task_helper module

```
class tamcolors.tests.test_multi_task_helper.MultiTaskHelper
    Bases: object

    multiple_processes_helper (tasks, timeout=120)
        info: will run tasks in a thread :param tasks: list or tuple: [(func, tuple, dict), ...] :param timeout: int
        :return:

    multiple_threads_helper (tasks, timeout=120)
        info: will run tasks in a thread :param tasks: list or tuple: [(func, tuple, dict), ...] :param timeout: int
        :return:

    classmethod task (func, *args, **kwargs)
        info: will make a Task :param func: Function :param args: tuple :param kwargs: dict :return: Task

tamcolors.tests.test_multi_task_helper.process_runner (error_ret, func, *args, **kwargs)
    info: will run the task :param error_ret: list: away to return an error :param func: function :param args: tuple
    :param kwargs: dict :return:
```

tamcolors.tests.test_utils module

```
tamcolors.tests.test_utils.enable_slow_tests (enable=True)
    info: will enable slow tests :param enable: bool :return:

tamcolors.tests.test_utils.is_slow_tests_enabled ()
    info: will check iif slow tests are enabled :return: bool

tamcolors.tests.test_utils.slow_test (function)
    info: slow_test decorators :param function: function :return: function
```

Module contents

tamcolors.utils package

Submodules

tamcolors.utils.compress module

```
tamcolors.utils.compress.compress(data)
    info: will compress data :param data: bytes :return: bytes

tamcolors.utils.compress.decompress(data)
    info: will defcompress data :param data: bytes :return: bytes
```

tamcolors.utils.encryption module

```
class tamcolors.utils.encryption.Encryption(rsa_key_size=4096, aes_key_size=256,
                                             nonce_key_size=64, authenticator=b'',
                                             max_random=20)

    Bases: object

    static check_sandy_check_key(sandy_key, data)
        info: Checks if the sandy key is equal to the data key. :param sandy_key: bytes: sha hash key :param data:
        bytes: :return: bool

    decrypt(data)
        info: Decrypt message. The data will also be checked for corruption. :param data: bytes :return: bytes:
        Unencrypted data.

    encrypt(data)
        info: Encrypts data. :param data: bytes :return: bytes: Encrypted data.

    encrypt_with_public_key(key, data)
        info: Encrypts data with public key. :param key: bytes or ras key object :param data: bytes :return: bytes:
        encrypted data

    get_raw_private_key()
        info: Gets the raw private key. :return: bytes: PEM encoding

    get_raw_public_key()
        info: gets the raw public key. :return: bytes: PEM encoding

    static make_sandy_check_key(data)
        info: Hashes data. :param data: bytes :return: bytes: Hash key.

exception tamcolors.utils.encryption.EncryptionError
    Bases: Exception
```

tamcolors.utils.identifier module

```
tamcolors.utils.identifier.generate_identifier_bytes(identifier_file='/home/docs/checkouts/readthedocs.org/u
                                             identifier_size=500)

tamcolors.utils.identifier.get_identifier_bytes(identifier_file='/home/docs/checkouts/readthedocs.org/user_bu
```

tamcolors.utils.immutable_cache module

```
class tamcolors.utils.immutable_cache.ImmutableCache
    Bases: object
```

tamcolors.utils.log module

```
tamcolors.utils.log.critical (*args, **kwargs)
    info: log at debug level :param args: tuple :param kwargs: dict :return:

tamcolors.utils.log.debug (*args, **kwargs)
    info: log at debug level :param args: tuple :param kwargs: dict :return:

tamcolors.utils.log.disable_logging()
    info: will disable logging :return:

tamcolors.utils.log.enable_logging (level=10)
    info: will enable logging at a level :param level: log level :return:

tamcolors.utils.log.error (*args, **kwargs)
    info: log at debug level :param args: tuple :param kwargs: dict :return:

tamcolors.utils.log.info (*args, **kwargs)
    info: log at info level :param args: tuple :param kwargs: dict :return:

tamcolors.utils.log.warning (*args, **kwargs)
    info: log at warning level :param args: tuple :param kwargs: dict :return:
```

tamcolors.utils.object_packer module

```
class tamcolors.utils.object_packer.FastHandObjectPacker
    Bases: object

    classmethod from_bytes (object_byte_array)
        info: from bytes to object :param object_byte_array: bytearray :return: object

    classmethod start_from_bytes (object_bytes)
        info: from bytes to object :param object_bytes: bytearray, bytes, list, tuple :return: object

    to_bytes ()
        info: object to bytes :return: bytes

class tamcolors.utils.object_packer.ObjectPackerJson (fast_hand_object_packer_objects=None)
    Bases: object

    dumps (data)
        info: object to bytes :param data: object :return: bytes

    loads (data)
        info: bytes to object :param data: bytearray, bytes, list, tuple :return: object

exception tamcolors.utils.object_packer.ObjectPackerJsonError
    Bases: Exception

tamcolors.utils.object_packer.load_data (object_byte_array)
    info: Loads bytes :param object_byte_array: bytearray :return: bytes

tamcolors.utils.object_packer.load_int (object_byte_array)
    info: Loads an unsigned int :param object_byte_array: bytearray :return: int

tamcolors.utils.object_packer.save_data
    info: Saves bytes :param data: bytes :return: bytes

tamcolors.utils.object_packer.save_int
    info: Saves an unsigned int :param number: int :return: bytes
```

tamcolors.utils.tcp module

```

class tamcolors.utils.tcp.TCPBase(connection, address, port, connection_password, encryption=None, object_packer=None, our_information=None)
    Bases: object

    close()
        info: will close the connection :return: None

    get_address()
        info: will get the address :return: str

    get_data()
        info: will compress and encrypt data if encryption is enabled :return: bytes

    is_open()
        info: will check if connection is open :return: bool

    send_data(data)
        info: will compress and decrypt if encryption is enabled :param data: bytes :return: None

class tamcolors.utils.tcp.TCPConnection(host='127.0.0.1', port=4444, ipv6=False, connection_password="", user_name=None, user_id=None, encryption=None, object_packer=None, our_information=None)
    Bases: tamcolors.utils.tcp.TCPBase

    get_other_data()
        info: will get other data :return: object

exception tamcolors.utils.tcp.TCPError
    Bases: Exception

class tamcolors.utils.tcp.TCPHost(*args, **kwargs)
    Bases: tamcolors.utils.tcp.TCPBase

    get_other_data()
        info: will get other data :return: object

    get_user_id()
        info: get connection id :return: bytes

    get_user_name()
        info: get connection name :return: str

class tamcolors.utils.tcp.TCPObjectConnector(tcp_connection, object_packer=None, no_return=None, optimizer=None)
    Bases: object

    close()
        info: will close the object :return:

    get_connection()
        info: will get connection :return: TCPBase

    is_open()
        info: will check if object is still open :return: bool

class tamcolors.utils.tcp.TCPObjectWrapper(tcp_connection, obj, object_packer=None)
    Bases: object

    close()
        info: will close the object :return:

```

```
get_connection()
    info: will get connection :return: TCPBase

is_open()
    info: will check if object is still open :return: bool

class tamcolors.utils.tcp.TCPReceiver(host='127.0.0.1', port=4444, ipv6=False, lis-
                                     ten_count=10, connection_password="", ad-
                                     dress_white_list=None, address_black_list=None,
                                     encryption=None, object_packer=None,
                                     our_information=None)

Bases: object

close()
    info: will close tcp receiver connection :return: None

get_host_connection(wait=True)
    info: will get a new host connection :param wait: bool: if true will wait for a connection :return: None or
    TCPHost
```

tamcolors.utils.timer module

```
class tamcolors.utils.timer.Timer(time_corruption=0)
    Bases: object

lap()
    Gets time difference from last lap. :return: float

offset_sleep(sleep_time)
    Will sleep the thread for a length of time based of the lap time. :param sleep_time: float :return: float
```

tamcolors.utils.transport_optimizer module

```
class tamcolors.utils.transport_optimizer.LastReceivedCache
    Bases: object

class tamcolors.utils.transport_optimizer.LastSentCache
    Bases: object
```

Module contents

Module contents

```
tamcolors.printc(*output, same_color=False, sep=' ', end='\n', flush=True, stderr=False)
    info: Prints color output to the console :param output: :param same_color: bool :param sep: str :param end: str
    :param flush: bool :param stderr: std :return: None

tamcolors.inputc(output, color)
    info: Will get color input from console :param output: str :param color: tuple :return: str

tamcolors.clear()
    info: Clears the console :return: None
```


CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

t

tamcolors, 60

tamcolors.examples, 17

tamcolors.examples.alpha, 13

tamcolors.examples.basic, 14

tamcolors.examples.clouds, 14

tamcolors.examples.colors, 14

tamcolors.examples.connection_loopback, 14

tamcolors.examples.host_loopback, 14

tamcolors.examples.icon, 14

tamcolors.examples.rgb_color, 15

tamcolors.examples.sandy_check, 15

tamcolors.examples.tabletennis, 15

tamcolors.examples.tam_key_manager, 15

tamcolors.examples.tam_list_buffer, 16

tamcolors.examples.tam_loop, 16

tamcolors.examples.tam_print, 16

tamcolors.examples.tam_text_box, 16

tamcolors.tam, 18

tamcolors.tam.tam_loop, 17

tamcolors.tam_basic, 19

tamcolors.tam_basic.basic, 19

tamcolors.tam_c, 19

tamcolors.tam_io, 31

tamcolors.tam_io.ansi_256_drivers, 19

tamcolors.tam_io.ansi_true_color_drivers, 20

tamcolors.tam_io.any_drivers, 20

tamcolors.tam_io.io_tam, 21

tamcolors.tam_io.tam_buffer, 24

tamcolors.tam_io.tam_colors, 25

tamcolors.tam_io.tam_drivers, 26

tamcolors.tam_io.tam_identifier, 29

tamcolors.tam_io.tam_keys, 29

tamcolors.tam_io.tcp_io, 29

tamcolors.tam_io.uni_drivers, 29

tamcolors.tam_io.win_drivers, 30

tamcolors.tam_tools, 38

tamcolors.tam_tools.tam_color_palette, 31

tamcolors.tam_tools.tam_fade, 32

tamcolors.tam_tools.tam_film, 33

tamcolors.tam_tools.tam_icon, 33

tamcolors.tam_tools.tam_key_manager, 34

tamcolors.tam_tools.tam_list_buffer, 35

tamcolors.tam_tools.tam_menu, 35

tamcolors.tam_tools.tam_placing, 37

tamcolors.tam_tools.tam_print, 37

tamcolors.tam_tools.tam_str, 38

tamcolors.tam_tools.tam_text_box, 38

tamcolors.tests, 56

tamcolors.tests.all_tests, 56

tamcolors.tests.tam_basic_tests, 39

tamcolors.tests.tam_basic_tests.basic_tests, 39

tamcolors.tests.tam_io_tests, 44

tamcolors.tests.tam_io_tests.any_drivers_tests, 39

tamcolors.tests.tam_io_tests.io_tam_tests, 40

tamcolors.tests.tam_io_tests.tam_buffer_tests, 40

tamcolors.tests.tam_io_tests.tam_colors_tests, 42

tamcolors.tests.tam_io_tests.tam_keys_tests, 42

tamcolors.tests.tam_io_tests.tam_standard_tests, 43

tamcolors.tests.tam_io_tests.uni_drivers_tests, 43

tamcolors.tests.tam_io_tests.win_drivers_tests, 43

tamcolors.tests.tam_tests, 45

tamcolors.tests.tam_tests.tam_loop_tests, 44

tamcolors.tests.tam_tools_tests, 52

tamcolors.tests.tam_tools_tests.tam_color_palette_t

tamcolors.tests.tam_tools_tests.tam_fade_tests,
46
tamcolors.tests.tam_tools_tests.tam_film_tests,
47
tamcolors.tests.tam_tools_tests.tam_key_manager_tests,
48
tamcolors.tests.tam_tools_tests.tam_list_buffer_tests,
48
tamcolors.tests.tam_tools_tests.tam_menu_tests,
49
tamcolors.tests.tam_tools_tests.tam_placing_tests,
50
tamcolors.tests.tam_tools_tests.tam_print_tests,
51
tamcolors.tests.tam_tools_tests.tam_str_tests,
51
tamcolors.tests.tam_tools_tests.tam_text_box_tests,
51
tamcolors.tests.test_multi_task_helper,
56
tamcolors.tests.test_utils, 56
tamcolors.tests.tests_tests, 53
tamcolors.tests.tests_tests.test_multi_task_helper_tests,
52
tamcolors.tests.utils_tests, 56
tamcolors.tests.utils_tests.compress_tests,
53
tamcolors.tests.utils_tests.encryption_tests,
53
tamcolors.tests.utils_tests.identifier_tests,
53
tamcolors.tests.utils_tests.immutable_cache_tests,
53
tamcolors.tests.utils_tests.log_tests,
54
tamcolors.tests.utils_tests.object_packer_tests,
54
tamcolors.tests.utils_tests.tcp_tests,
55
tamcolors.tests.utils_tests.timer_tests,
55
tamcolors.tests.utils_tests.transport_optimizer_tests,
55
tamcolors.utils, 60
tamcolors.utils.compress, 57
tamcolors.utils.encryption, 57
tamcolors.utils.identifier, 57
tamcolors.utils.immutable_cache, 57
tamcolors.utils.log, 58
tamcolors.utils.object_packer, 58
tamcolors.utils.tcp, 59
tamcolors.utils.timer, 60
tamcolors.utils.transport_optimizer, 60

A

[a](#) (*tamcolors.tam_io.tam_colors.RGBA* attribute), 26
[able_to_execute\(\)](#) (*tamcolors.tam_io.io_tam.IO* class method), 21
[able_to_execute\(\)](#) (*tamcolors.tam_io.io_tam.RawIO* class method), 22
[able_to_execute\(\)](#) (*tamcolors.tam_io.tam_drivers.TAMDriver* class method), 28
[able_to_execute\(\)](#) (*tamcolors.tam_io.uni_drivers.UNISharedData* class method), 30
[able_to_execute\(\)](#) (*tamcolors.tam_io.win_drivers.WinSharedData* class method), 31
[add_frame_stack\(\)](#) (*tamcolors.tam.tam_loop.TAMLoop* method), 17
[ANSI256ChangerDriver](#) (class in *tamcolors.tam_io.ansi_256_drivers*), 19
[ANSI256ColorDriver](#) (class in *tamcolors.tam_io.ansi_256_drivers*), 19
[ANSITrueFullColorDriver](#) (class in *tamcolors.tam_io.ansi_true_color_drivers*), 20
[ANYColorChangerDriver](#) (class in *tamcolors.tam_io.any_drivers*), 20
[ANYColorDriver](#) (class in *tamcolors.tam_io.any_drivers*), 20
[AnyIOTests](#) (class in *tamcolors.tests.tam_io_tests.any_drivers_tests*), 39
[ANYKeyDriver](#) (class in *tamcolors.tam_io.any_drivers*), 20
[ANYUtilitiesDriver](#) (class in *tamcolors.tam_io.any_drivers*), 20
[append\(\)](#) (*tamcolors.tam_tools.tam_film.TAMFilm* method), 33

B

[b](#) (*tamcolors.tam_io.tam_colors.RGBA* attribute), 26
[Ball](#) (class in *tamcolors.examples.tabletennis*), 15
[BootLogo](#) (class in *tamcolors.examples.icon*), 14
[bottom_left\(\)](#) (in module *tamcolors.tam_tools.tam_placing*), 37
[bottom_right\(\)](#) (in module *tamcolors.tam_tools.tam_placing*), 37

C

[center\(\)](#) (in module *tamcolors.tam_tools.tam_placing*), 37
[check_sandy_check_key\(\)](#) (*tamcolors.utils.encryption.Encryption* static method), 57
[clear\(\)](#) (in module *tamcolors*), 60
[clear\(\)](#) (in module *tamcolors.tam_basic.basic*), 19
[clear\(\)](#) (*tamcolors.tam_io.any_drivers.ANYUtilitiesDriver* method), 20
[clear\(\)](#) (*tamcolors.tam_io.io_tam.IO* method), 21
[clear\(\)](#) (*tamcolors.tam_io.io_tam.RawIO* method), 22
[clear\(\)](#) (*tamcolors.tam_io.tam_buffer.TAMBuffer* method), 24
[clear\(\)](#) (*tamcolors.tam_io.tam_drivers.UtilitiesDriver* method), 28
[clear\(\)](#) (*tamcolors.tam_io.uni_drivers.UNIUtilitiesDriver* method), 30
[clear\(\)](#) (*tamcolors.tam_io.win_drivers.WINUtilitiesDriver* method), 31
[ClearTests](#) (class in *tamcolors.tests.tam_basic_tests.basic_tests*), 39
[close\(\)](#) (*tamcolors.utils.tcp.TCPBase* method), 59
[close\(\)](#) (*tamcolors.utils.tcp.TCPObjectConnector* method), 59
[close\(\)](#) (*tamcolors.utils.tcp.TCPObjectWrapper* method), 59
[close\(\)](#) (*tamcolors.utils.tcp.TCPReceiver* method), 60
[Clouds](#) (class in *tamcolors.examples.clouds*), 14
[Color](#) (class in *tamcolors.tam_io.tam_colors*), 25

`color_change_driver_operational()` (*tamcolors.tam_io.io_tam.RawIO method*), 22
`color_changer_driver_operational()` (*tamcolors.tam_io.io_tam.IO method*), 21
`color_changer_driver_operational()` (*tamcolors.tam_io.io_tam.RawIO method*), 23
`color_driver_operational()` (*tamcolors.tam_io.io_tam.IO method*), 21
`color_driver_operational()` (*tamcolors.tam_io.io_tam.RawIO method*), 23
`ColorChangerDriver` (class in *tamcolors.tam_io.tam_drivers*), 26
`ColorDriver` (class in *tamcolors.tam_io.tam_drivers*), 27
`ColorNameToCodeTests` (class in *tamcolors.tests.tam_basic_tests.basic_tests*), 39
`ColorTests` (class in *tamcolors.tests.tam_io_tests.tam_colors_tests*), 42
`compress()` (in module *tamcolors.utils.compress*), 57
`CompressTests` (class in *tamcolors.tests.utils_tests.compress_tests*), 53
`copy()` (*tamcolors.tam_io.tam_buffer.TAMBuffer method*), 24
`critical()` (in module *tamcolors.utils.log*), 58

D

`debug()` (in module *tamcolors.utils.log*), 58
`decompress()` (in module *tamcolors.utils.compress*), 57
`decrypt()` (*tamcolors.utils.encryption.Encryption method*), 57
`disable_logging()` (in module *tamcolors.utils.log*), 58
`done()` (*tamcolors.tam.tam_loop.TAMFrame method*), 17
`done()` (*tamcolors.tam.tam_loop.TAMLoop method*), 17
`done()` (*tamcolors.tam_io.io_tam.IO method*), 21
`done()` (*tamcolors.tam_io.io_tam.RawIO method*), 23
`done()` (*tamcolors.tam_io.tam_drivers.TAMDriver method*), 28
`done()` (*tamcolors.tam_io.uni_drivers.UNIUtilitiesDriver method*), 30
`done()` (*tamcolors.tam_io.win_drivers.WINFullColorDriver method*), 30
`done()` (*tamcolors.tam_io.win_drivers.WINUtilitiesDriver method*), 31
`done()` (*tamcolors.tam_tools.tam_film.TAMFilm method*), 33
`done()` (*tamcolors.tam_tools.tam_text_box.TAMTextBox method*), 38
`draw()` (*tamcolors.examples.alpha.TAMAlpha method*), 13
`draw()` (*tamcolors.examples.clouds.Clouds method*), 14
`draw()` (*tamcolors.examples.colors.TAMCOLORS method*), 14
`draw()` (*tamcolors.examples.icon.BootLogo method*), 14
`draw()` (*tamcolors.examples.rgb_color.RGBCOLOR method*), 15
`draw()` (*tamcolors.examples.tabletennis.TableTennis method*), 15
`draw()` (*tamcolors.examples.tam_key_manager.TAMKeyManager method*), 15
`draw()` (*tamcolors.examples.tam_list_buffer.TAMListBuffer method*), 16
`draw()` (*tamcolors.examples.tam_loop.TAMLoopHelloWorld method*), 16
`draw()` (*tamcolors.examples.tam_print.TAMPrint method*), 16
`draw()` (*tamcolors.examples.tam_text_box.TAMPrint method*), 16
`draw()` (*tamcolors.tam.tam_loop.TAMFrame method*), 17
`draw()` (*tamcolors.tam_io.ansi_256_drivers.ANSI256ColorDriver method*), 19
`draw()` (*tamcolors.tam_io.ansi_true_color_drivers.ANSITrueFullColorDriver method*), 20
`draw()` (*tamcolors.tam_io.any_drivers.ANYColorDriver method*), 20
`draw()` (*tamcolors.tam_io.io_tam.IO method*), 21
`draw()` (*tamcolors.tam_io.io_tam.RawIO method*), 23
`draw()` (*tamcolors.tam_io.tam_drivers.ColorDriver method*), 27
`draw()` (*tamcolors.tam_io.win_drivers.WINFullColorDriver method*), 30
`draw()` (*tamcolors.tam_tools.tam_menu.TAMButtonRule method*), 35
`draw()` (*tamcolors.tam_tools.tam_menu.TAMMenu method*), 35
`draw()` (*tamcolors.tam_tools.tam_menu.TAMTextBoxButton method*), 36
`draw()` (*tamcolors.tam_tools.tam_menu.TAMTextButton method*), 36
`draw()` (*tamcolors.tam_tools.tam_text_box.TAMTextBox method*), 38
`draw_onto()` (*tamcolors.tam_io.tam_buffer.TAMBuffer method*), 24
`DummyCache` (class in *tamcolors.tests.utils_tests.immutable_cache_tests*), 53
`dumps()` (*tamcolors.utils.object_packer.ObjectPackerJson method*), 58

E

`enable_console_keys()` (*tamcol-*

ors.tam_io.io_tam.IO method), 21
 enable_console_keys() (*tamcol-ors.tam_io.io_tam.RawIO method*), 23
 enable_console_keys() (*tamcol-ors.tam_io.tam_drivers.KeyDriver method*), 28
 enable_console_keys() (*tamcol-ors.tam_io.uni_drivers.UNIKeyDriver method*), 29
 enable_logging() (*in module tamcolors.utils.log*), 58
 enable_slow_tests() (*in module tamcol-ors.tests.test_utils*), 56
 encrypt() (*tamcolors.utils.encryption.Encryption method*), 57
 encrypt_with_public_key() (*tamcol-ors.utils.encryption.Encryption method*), 57
 Encryption (*class in tamcolors.utils.encryption*), 57
 EncryptionError, 57
 EncryptionTests (*class in tamcol-ors.tests.utils_tests.encryption_tests*), 53
 error() (*in module tamcolors.utils.log*), 58

F

FastHandObjectPacker (*class in tamcol-ors.utils.object_packer*), 58
 FastHandObjectPackerTests (*class in tamcol-ors.tests.utils_tests.object_packer_tests*), 54
 from_bytes() (*tamcol-ors.tam_io.tam_buffer.TAMBuffer class method*), 24
 from_bytes() (*tamcolors.tam_io.tam_colors.Color class method*), 25
 from_bytes() (*tamcolors.tam_io.tam_colors.RGBA class method*), 26
 from_bytes() (*tamcol-ors.utils.object_packer.FastHandObjectPacker class method*), 58
 FullColorDriver (*class in tamcol-ors.tam_io.tam_drivers*), 27

G

g (*tamcolors.tam_io.tam_colors.RGBA attribute*), 26
 generate_identifier_bytes() (*in module tamcolors.utils.identifier*), 57
 get() (*tamcolors.tam_tools.tam_film.TAMFilm method*), 33
 get_action() (*tamcol-ors.tam_tools.tam_menu.TAMButtonRule method*), 35
 get_action() (*tamcol-ors.tam_tools.tam_menu.TAMTextBoxButton method*), 36
 get_action() (*tamcol-ors.tam_tools.tam_menu.TAMTextButton method*), 36
 get_address() (*tamcolors.utils.tcp.TCPBase method*), 59
 get_all_drivers() (*tamcol-ors.tam_io.tam_identifier.TAMIdentifier method*), 29
 get_any_io() (*in module tamcol-ors.tests.tam_io_tests.any_drivers_tests*), 40
 get_buttons() (*tamcol-ors.tam_tools.tam_menu.TAMMenu method*), 35
 get_call_key() (*tamcol-ors.tam_tools.tam_menu.TAMMenu method*), 35
 get_char() (*tamcol-ors.tam_tools.tam_text_box.TAMTextBox method*), 38
 get_circular() (*tamcol-ors.tam_tools.tam_film.TAMFilm method*), 33
 get_clock() (*tamcol-ors.tam_tools.tam_color_palette.TAMCycleColor method*), 32
 get_color() (*tamcol-ors.tam_tools.tam_color_palette.TAMColorPalette method*), 31
 get_color() (*tamcol-ors.tam_tools.tam_color_palette.TAMDefaultColor method*), 32
 get_color_16() (*tamcolors.tam.tam_loop.TAMLoop method*), 17
 get_color_16() (*tamcolors.tam_io.io_tam.IO method*), 21
 get_color_16() (*tamcolors.tam_io.io_tam.RawIO method*), 23
 get_color_16() (*tamcol-ors.tam_io.tam_drivers.ColorChangerDriver method*), 26
 get_color_16_pal_256() (*tamcol-ors.tam.tam_loop.TAMLoop method*), 17
 get_color_16_pal_256() (*tamcol-ors.tam_io.io_tam.IO method*), 21
 get_color_16_pal_256() (*tamcol-ors.tam_io.io_tam.RawIO method*), 23
 get_color_16_pal_256() (*tamcol-ors.tam_io.tam_drivers.ColorChangerDriver method*), 26
 get_color_2() (*tamcolors.tam.tam_loop.TAMLoop method*), 18
 get_color_2() (*tamcolors.tam_io.io_tam.IO method*), 21

<code>get_color_2()</code> (<i>tamcolors.tam_io.io_tam.RawIO method</i>), 23	<code>get_fps()</code> (<i>tamcolors.tam.tam_loop.TAMFrame method</i>), 17
<code>get_color_2()</code> (<i>tamcolors.tam_io.tam_drivers.ColorChangerDriver method</i>), 26	<code>get_from_raw_spot()</code> (<i>tamcolors.tam_io.tam_buffer.TAMBuffer method</i>), 25
<code>get_color_256()</code> (<i>tamcolors.tam.tam_loop.TAMLoop method</i>), 18	<code>get_goto_map()</code> (<i>tamcolors.tam_tools.tam_menu.TAMMenu method</i>), 35
<code>get_color_256()</code> (<i>tamcolors.tam_io.io_tam.IO method</i>), 21	<code>get_height_min_and_max()</code> (<i>tamcolors.tam.tam_loop.TAMFrame method</i>), 17
<code>get_color_256()</code> (<i>tamcolors.tam_io.io_tam.RawIO method</i>), 23	<code>get_host_connection()</code> (<i>tamcolors.utils.tcp.TCPReceiver method</i>), 60
<code>get_color_256()</code> (<i>tamcolors.tam_io.tam_drivers.ColorChangerDriver method</i>), 26	<code>get_icon()</code> (<i>in module tamcolors.tam_tools.tam_icon</i>), 33
<code>get_colors()</code> (<i>tamcolors.tam_tools.tam_text_box.TAMTextBox method</i>), 38	<code>get_identifier_bytes()</code> (<i>in module tamcolors.utils.identifier</i>), 57
<code>get_connection()</code> (<i>tamcolors.utils.tcp.TCPObjectConnector method</i>), 59	<code>get_info_dict()</code> (<i>tamcolors.tam_io.io_tam.IO method</i>), 21
<code>get_connection()</code> (<i>tamcolors.utils.tcp.TCPObjectWrapper method</i>), 59	<code>get_info_dict()</code> (<i>tamcolors.tam_io.io_tam.RawIO method</i>), 23
<code>get_cross_rect()</code> (<i>tamcolors.tam_io.tam_buffer.TAMBuffer method</i>), 24	<code>get_info_dict()</code> (<i>tamcolors.tam_io.tam_identifier.TAMIdentifier method</i>), 29
<code>get_data()</code> (<i>tamcolors.utils.tcp.TCPBase method</i>), 59	<code>get_io()</code> (<i>tamcolors.tam_io.tam_identifier.TAMIdentifier method</i>), 29
<code>get_defaults()</code> (<i>tamcolors.tam.tam_loop.TAMFrame method</i>), 17	<code>get_key()</code> (<i>tamcolors.tam_io.any_drivers.ANYKeyDriver method</i>), 20
<code>get_defaults()</code> (<i>tamcolors.tam_io.tam_buffer.TAMBuffer method</i>), 24	<code>get_key()</code> (<i>tamcolors.tam_io.io_tam.IO method</i>), 21
<code>get_dimensions()</code> (<i>tamcolors.tam_io.any_drivers.ANYUtilitiesDriver method</i>), 20	<code>get_key()</code> (<i>tamcolors.tam_io.io_tam.RawIO method</i>), 23
<code>get_dimensions()</code> (<i>tamcolors.tam_io.io_tam.IO method</i>), 21	<code>get_key()</code> (<i>tamcolors.tam_io.tam_drivers.KeyDriver method</i>), 28
<code>get_dimensions()</code> (<i>tamcolors.tam_io.io_tam.RawIO method</i>), 23	<code>get_key()</code> (<i>tamcolors.tam_io.uni_drivers.UNIKeyDriver method</i>), 29
<code>get_dimensions()</code> (<i>tamcolors.tam_io.tam_buffer.TAMBuffer method</i>), 25	<code>get_key()</code> (<i>tamcolors.tam_io.win_drivers.WINKeyDriver method</i>), 30
<code>get_dimensions()</code> (<i>tamcolors.tam_io.tam_drivers.UtilitiesDriver method</i>), 28	<code>get_key_dict()</code> (<i>tamcolors.tam_io.any_drivers.ANYKeyDriver static method</i>), 20
<code>get_dimensions()</code> (<i>tamcolors.tam_io.uni_drivers.UNIUtilitiesDriver method</i>), 30	<code>get_key_dict()</code> (<i>tamcolors.tam_io.io_tam.IO static method</i>), 21
<code>get_dimensions()</code> (<i>tamcolors.tam_io.win_drivers.WINUtilitiesDriver method</i>), 31	<code>get_key_dict()</code> (<i>tamcolors.tam_io.io_tam.RawIO static method</i>), 23
<code>get_dimensions()</code> (<i>tamcolors.tam_tools.tam_text_box.TAMTextBox method</i>), 38	<code>get_key_dict()</code> (<i>tamcolors.tam_io.tam_drivers.KeyDriver static method</i>), 28
	<code>get_key_dict()</code> (<i>tamcolors.tam_io.uni_drivers.UNIKeyDriver static method</i>), 29
	<code>get_key_dict()</code> (<i>tamcolors.tam_io.win_drivers.WINKeyDriver static method</i>), 31
	<code>get_key_state()</code> (<i>tamcolors.tam_tools.tam_key_manager.TAMKeyManager</i>

`method`), 34
`get_mode()` (*tamcolors.tam_io.io_tam.IO method*), 21
`get_mode()` (*tamcolors.tam_io.io_tam.RawIO method*), 23
`get_modes()` (*tamcolors.tam_io.io_tam.IO method*), 22
`get_modes()` (*tamcolors.tam_io.io_tam.RawIO method*), 23
`get_name()` (*tamcolors.tam_io.tam_identifier.TAMIdentifier method*), 29
`get_on()` (*tamcolors.tam_tools.tam_menu.TAMMenu method*), 36
`get_other_data()` (*tamcolors.ors.utils.tcp.TCPConnection method*), 59
`get_other_data()` (*tamcolors.ors.utils.tcp.TCPHost method*), 59
`get_position()` (*tamcolors.tam_tools.tam_menu.TAMButtonRule method*), 35
`get_position()` (*tamcolors.tam_tools.tam_menu.TAMTextBoxButton method*), 36
`get_position()` (*tamcolors.tam_tools.tam_menu.TAMTextButton method*), 37
`get_printc_mode()` (*tamcolors.tam_io.ansi_256_drivers.ANSI256ColorDriver method*), 19
`get_printc_mode()` (*tamcolors.tam_io.ansi_true_color_drivers.ANSITrueFullColorDriver method*), 20
`get_printc_mode()` (*tamcolors.tam_io.any_drivers.ANYColorDriver method*), 20
`get_printc_mode()` (*tamcolors.tam_io.io_tam.IO method*), 22
`get_printc_mode()` (*tamcolors.tam_io.io_tam.RawIO method*), 23
`get_printc_mode()` (*tamcolors.tam_io.tam_drivers.ColorDriver method*), 27
`get_printc_mode()` (*tamcolors.tam_io.win_drivers.WINFullColorDriver method*), 30
`get_raw_buffers()` (*tamcolors.tam_io.tam_buffer.TAMBuffer method*), 25
`get_raw_private_key()` (*tamcolors.ors.utils.encryption.Encryption method*), 57
`get_raw_public_key()` (*tamcolors.ors.utils.encryption.Encryption method*), 57
`get_raw_spot()` (*tamcolors.tam_io.tam_buffer.TAMBuffer method*), 25
`get_raw_user_input()` (*tamcolors.tam_tools.tam_key_manager.TAMKeyManager method*), 35
`get_raw_user_input_2()` (*tamcolors.ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManager method*), 48
`get_raw_user_input_3()` (*tamcolors.ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManager method*), 48
`get_rule()` (*tamcolors.tam_tools.tam_color_palette.TAMColorPalette method*), 31
`get_running()` (*tamcolors.tam.tam_loop.TAMLoop method*), 18
`get_spot()` (*tamcolors.tam_io.tam_buffer.TAMBuffer method*), 25
`get_system()` (*tamcolors.tam_io.tam_identifier.TAMIdentifier method*), 29
`get_tcp_io()` (*in module tamcolors.tam_io.tcp_io*), 29
`get_text()` (*tamcolors.tam_tools.tam_text_box.TAMTextBox method*), 38
`get_uni_io()` (*in module tamcolors.ors.tests.tam_io_tests.uni_drivers_tests*), 43
`get_color_driver_id()` (*tamcolors.ors.utils.tcp.TCPHost method*), 59
`get_user_input()` (*tamcolors.tam_tools.tam_key_manager.TAMKeyManager method*), 35
`get_user_input_generator()` (*tamcolors.tam_tools.tam_key_manager.TAMKeyManager method*), 35
`get_user_name()` (*tamcolors.ors.utils.tcp.TCPHost method*), 59
`get_width_min_and_max()` (*tamcolors.tam.tam_loop.TAMFrame method*), 17
`get_win_io()` (*in module tamcolors.ors.tests.tam_io_tests.win_drivers_tests*), 44
`get_y()` (*tamcolors.examples.tabletennis.Ball method*), 15
`GetColorCodeTests` (*class in tamcolors.ors.tests.tam_basic_tests.basic_tests*), 39

H

`has_alpha` (*tamcolors.tam_io.tam_colors.Color attribute*), 25

I

IdentifierTests (class in tamcolors.tests.utils_tests.identifier_tests), 53

identify() (tamcolors.tam_io.tam_identifier.TAMIdentifier class method), 29

ImmutableCache (class in tamcolors.utils.immutable_cache), 57

ImmutableCacheTests (class in tamcolors.tests.utils_tests.immutable_cache_tests), 53

info() (in module tamcolors.utils.log), 58

inputc() (in module tamcolors), 60

inputc() (in module tamcolors.tam_basic.basic), 19

inputc() (tamcolors.tam_io.ansi_256_drivers.ANSI256ColorDriver method), 19

inputc() (tamcolors.tam_io.ansi_true_color_drivers.ANSITruFullColorDriver method), 20

inputc() (tamcolors.tam_io.any_drivers.ANYColorDriver method), 20

inputc() (tamcolors.tam_io.io_tam.IO method), 22

inputc() (tamcolors.tam_io.io_tam.RawIO method), 23

inputc() (tamcolors.tam_io.tam_drivers.ColorDriver method), 27

inputc() (tamcolors.tam_io.win_drivers.WINFullColorDriver method), 30

InputCTests (class in tamcolors.tests.tam_basic_tests.basic_tests), 39

IO (class in tamcolors.tam_io.io_tam), 21

IOTAMTest (class in tamcolors.tests.tam_io_tests.io_tam_tests), 40

IOTests (class in tamcolors.tests.tam_basic_tests.basic_tests), 39

is_console_cursor_enabled() (tamcolors.tam_io.io_tam.IO method), 22

is_console_cursor_enabled() (tamcolors.tam_io.io_tam.RawIO method), 23

is_console_keys_enabled() (tamcolors.tam_io.io_tam.IO method), 22

is_console_keys_enabled() (tamcolors.tam_io.io_tam.RawIO method), 23

is_default (tamcolors.tam_io.tam_colors.RGBA attribute), 26

is_open() (tamcolors.utils.tcp.TCPBase method), 59

is_open() (tamcolors.utils.tcp.TCPObjectConnector method), 59

is_open() (tamcolors.utils.tcp.TCPObjectWrapper method), 60

is_slow_tests_enabled() (in module tamcolors.tests.test_utils), 56

K

key_driver_operational() (tamcol-

ors.tam_io.io_tam.IO method), 22

key_driver_operational() (tamcol-ors.tam_io.io_tam.RawIO method), 23

key_present() (tamcol-ors.tam_tools.tam_color_palette.TAMColorPalette method), 31

KeyDriver (class in tamcolors.tam_io.tam_drivers), 27

L

lap() (tamcolors.utils.timer.Timer method), 60

LastReceivedCache (class in tamcolors.utils.transport_optimizer), 60

LastSentCache (class in tamcolors.utils.transport_optimizer), 60

LoadData() (in module tamcolors.utils.object_packer), 58

LoadFullColorDriver (in module tamcolors.utils.object_packer), 58

load_tests() (in module tamcolors.tests.all_tests), 56

loads() (tamcolors.utils.object_packer.ObjectPackerJson method), 58

LogTests (class in tamcolors.tests.utils_tests.log_tests), 54

M

make_buffer_ready() (tamcol-ors.tam.tam_loop.TAMFrame method), 17

make_sandy_check_key() (tamcol-ors.utils.encryption.Encryption static method), 57

make_tam_str() (in module tamcol-ors.tam_tools.tam_str), 38

MakeTAMStrTests (class in tamcolors.tests.tam_tools_tests.tam_str_tests), 51

mode_16 (tamcolors.tam_io.tam_colors.Color attribute), 25

mode_16_pal_256 (tamcol-ors.tam_io.tam_colors.Color attribute), 25

mode_2 (tamcolors.tam_io.tam_colors.Color attribute), 25

mode_256 (tamcolors.tam_io.tam_colors.Color attribute), 25

mode_rgb (tamcolors.tam_io.tam_colors.Color attribute), 25

multiple_processes_helper() (tamcol-ors.tests.test_multi_task_helper.MultiTaskHelper method), 56

multiple_threads_helper() (tamcol-ors.tests.test_multi_task_helper.MultiTaskHelper method), 56

MultiTaskHelper (class in tamcol-ors.tests.test_multi_task_helper), 56

MultiTaskHelperTests (class in tamcol-
ors.tests.tests_tests.test_multi_task_helper_tests),
52

N

NoEncryptionTests (class in tamcol-
ors.tests.utils_tests.encryption_tests), 53

O

ObjectPackerFunctionsTests (class in tamcol-
ors.tests.utils_tests.object_packer_tests), 54

ObjectPackerJson (class in tamcol-
ors.utils.object_packer), 58

ObjectPackerJsonError, 58

ObjectPackerJsonTests (class in tamcol-
ors.tests.utils_tests.object_packer_tests),
54

off() (tamcolors.tam_tools.tam_menu.TAMButtonRule
method), 35

off() (tamcolors.tam_tools.tam_menu.TAMTextBoxButton
method), 36

off() (tamcolors.tam_tools.tam_menu.TAMTextButton
method), 37

offset_sleep() (tamcolors.utils.timer.Timer
method), 60

on() (tamcolors.tam_tools.tam_menu.TAMButtonRule
method), 35

on() (tamcolors.tam_tools.tam_menu.TAMTextBoxButton
method), 36

on() (tamcolors.tam_tools.tam_menu.TAMTextButton
method), 37

P

peak() (tamcolors.tam_tools.tam_film.TAMFilm
method), 33

place_color_over() (tamcol-
ors.tam_io.tam_colors.Color method), 26

pop() (tamcolors.tam_tools.tam_film.TAMFilm
method), 33

pop_frame_stack() (tamcol-
ors.tam.tam_loop.TAMLoop method), 18

putc() (in module tamcolors), 60

putc() (in module tamcolors.tam_basic.basic), 19

putc() (tamcolors.tam_io.ansi_256_drivers.ANSI256ColorDriver
method), 19

putc() (tamcolors.tam_io.ansi_true_color_drivers.ANSITrueColorDriver
method), 20

putc() (tamcolors.tam_io.any_drivers.ANYColorDriver
method), 20

putc() (tamcolors.tam_io.io_tam.IO method), 22

putc() (tamcolors.tam_io.io_tam.RawIO method),
23

putc() (tamcolors.tam_io.tam_drivers.ColorDriver
method), 27

putc() (tamcolors.tam_io.win_drivers.WINFullColorDriver
method), 30

PrintCTests (class in tamcol-
ors.tests.tam_basic_tests.basic_tests), 39

process_runner() (in module tamcol-
ors.tests.test_multi_task_helper), 56

R

r (tamcolors.tam_io.tam_colors.RGBA attribute), 26

Racket (class in tamcolors.examples.tabletennis), 15

RawIO (class in tamcolors.tam_io.io_tam), 22

RAWIOTest (class in tamcol-
ors.tests.tam_io_tests.io_tam_tests), 40

replace_alpha_chars() (tamcol-
ors.tam_io.tam_buffer.TAMBuffer method),
25

reset_colors_to_console_defaults() (in
module tamcolors.tam_basic.basic), 19

reset_colors_to_console_defaults() (tam-
colors.tam.tam_loop.TAMLoop method), 18

reset_colors_to_console_defaults() (tam-
colors.tam_io.io_tam.IO method), 22

reset_colors_to_console_defaults() (tam-
colors.tam_io.io_tam.RawIO method), 24

RGBA (class in tamcolors.tam_io.tam_colors), 26

RGBATests (class in tamcol-
ors.tests.tam_io_tests.tam_colors_tests),
42

RGBCOLOR (class in tamcolors.examples.rgb_color), 15

run() (in module tamcolors.examples.alpha), 13

run() (in module tamcolors.examples.basic), 14

run() (in module tamcolors.examples.clouds), 14

run() (in module tamcolors.examples.colors), 14

run() (in module tamcol-
ors.examples.connection_loopback), 14

run() (in module tamcolors.examples.host_loopback),
14

run() (in module tamcolors.examples.icon), 14

run() (in module tamcolors.examples.rgb_color), 15

run() (in module tamcolors.examples.sandy_check), 15

run() (in module tamcolors.examples.tabletennis), 15

run() (in module tamcol-
ors.examples.tam_key_manager), 15

run() (in module tamcolors.examples.tam_list_buffer),
16

run() (in module tamcolors.examples.tam_loop), 16

run() (in module tamcolors.examples.tam_print), 16

run() (in module tamcolors.examples.tam_text_box), 16

run() (tamcolors.tam.tam_loop.TAMLoop method), 18

run_action() (tamcol-
ors.tam_tools.tam_menu.TAMButtonRule
method), 35

run_action() (tamcol-
ors.tam_tools.tam_menu.TAMTextBoxButton

method), 36
 run_action() (tamcol-
 ors.tam_tools.tam_menu.TAMTextButton
 method), 37
 run_application() (tamcol-
 ors.tam.tam_loop.TAMLoop static
 method), 18
 run_with_profiler() (tamcol-
 ors.tam.tam_loop.TAMLoop method), 18

S

save_data (in module tamcolors.utils.object_packer),
 58
 save_int (in module tamcolors.utils.object_packer),
 58
 send_data() (tamcolors.utils.tcp.TCPBase method),
 59
 set() (tamcolors.tam_tools.tam_film.TAMFilm
 method), 33
 set_action() (tamcol-
 ors.tam_tools.tam_menu.TAMButtonRule
 method), 35
 set_action() (tamcol-
 ors.tam_tools.tam_menu.TAMTextBoxButton
 method), 36
 set_action() (tamcol-
 ors.tam_tools.tam_menu.TAMTextButton
 method), 37
 set_char() (tamcol-
 ors.tam_tools.tam_text_box.TAMTextBox
 method), 38
 set_circular() (tamcol-
 ors.tam_tools.tam_film.TAMFilm method),
 33
 set_clock() (tamcol-
 ors.tam_tools.tam_color_palette.TAMCycleColor
 method), 32
 set_color() (tamcol-
 ors.tam_tools.tam_color_palette.TAMColorPalette
 method), 32
 set_color() (tamcol-
 ors.tam_tools.tam_color_palette.TAMDefaultColor
 method), 32
 set_color_16() (tamcolors.tam.tam_loop.TAMLoop
 method), 18
 set_color_16() (tamcolors.tam_io.io_tam.IO
 method), 22
 set_color_16() (tamcolors.tam_io.io_tam.RawIO
 method), 24
 set_color_16() (tamcol-
 ors.tam_io.tam_drivers.ColorChangerDriver
 method), 27
 set_color_16_pal_256() (tamcol-
 ors.tam.tam_loop.TAMLoop method), 18
 set_color_16_pal_256() (tamcol-
 ors.tam_io.io_tam.IO method), 22
 set_color_16_pal_256() (tamcol-
 ors.tam_io.io_tam.RawIO method), 24
 set_color_16_pal_256() (tamcol-
 ors.tam_io.tam_drivers.ColorChangerDriver
 method), 27
 set_color_2() (tamcolors.tam.tam_loop.TAMLoop
 method), 18
 set_color_2() (tamcolors.tam_io.io_tam.IO
 method), 22
 set_color_2() (tamcolors.tam_io.io_tam.RawIO
 method), 24
 set_color_2() (tamcol-
 ors.tam_io.tam_drivers.ColorChangerDriver
 method), 27
 set_color_256() (tamcol-
 ors.tam.tam_loop.TAMLoop method), 18
 set_color_256() (tamcolors.tam_io.io_tam.IO
 method), 22
 set_color_256() (tamcolors.tam_io.io_tam.RawIO
 method), 24
 set_color_256() (tamcol-
 ors.tam_io.tam_drivers.ColorChangerDriver
 method), 27
 set_colors() (tamcol-
 ors.tam_tools.tam_color_palette.TAMCycleColor
 method), 32
 set_colors() (tamcol-
 ors.tam_tools.tam_text_box.TAMTextBox
 method), 38
 set_defaults_and_clear() (tamcol-
 ors.tam_io.tam_buffer.TAMBuffer method),
 25
 set_dimensions_and_clear() (tamcol-
 ors.tam_io.tam_buffer.TAMBuffer method),
 25
 set_mode() (tamcol-
 ors.tam_io.ansi_256_drivers.ANSI256ColorDriver
 method), 19
 set_mode() (tamcol-
 ors.tam_io.ansi_true_color_drivers.ANSITrueFull
 ColorDriver method), 20
 set_mode() (tamcolors.tam_io.io_tam.IO method), 22
 set_mode() (tamcolors.tam_io.io_tam.RawIO
 method), 24
 set_mode() (tamcol-
 ors.tam_io.tam_drivers.ColorChangerDriver
 method), 27
 set_mode() (tamcol-
 ors.tam_io.tam_drivers.ColorDriver method),
 27
 set_mode() (tamcol-
 ors.tam_io.win_drivers.WINFullColorDriver

method), 30
 set_position() (tamcolors.ors.tam_tools.tam_menu.TAMButtonRule method), 35
 set_position() (tamcolors.ors.tam_tools.tam_menu.TAMTextBoxButton method), 36
 set_position() (tamcolors.ors.tam_tools.tam_menu.TAMTextButton method), 37
 set_rule() (tamcolors.ors.tam_tools.tam_color_palette.TAMColorPalette method), 32
 set_spot() (tamcolors.tam_io.tam_buffer.TAMBuffer method), 25
 set_tam_color_defaults() (in module tamcolors.ors.tam_basic.basic), 19
 set_tam_color_defaults() (tamcolors.ors.tam.tam_loop.TAMLoop method), 18
 set_tam_color_defaults() (tamcolors.ors.tam_io.io_tam.IO method), 22
 set_tam_color_defaults() (tamcolors.ors.tam_io.io_tam.RawIO method), 24
 show_console_cursor() (tamcolors.ors.tam_io.any_drivers.ANYUtilitiesDriver method), 21
 show_console_cursor() (tamcolors.ors.tam_io.io_tam.IO method), 22
 show_console_cursor() (tamcolors.ors.tam_io.io_tam.RawIO method), 24
 show_console_cursor() (tamcolors.ors.tam_io.tam_drivers.UtilitiesDriver method), 28
 show_console_cursor() (tamcolors.ors.tam_io.uni_drivers.UNIUtilitiesDriver method), 30
 show_console_cursor() (tamcolors.ors.tam_io.win_drivers.WINUtilitiesDriver method), 31
 silent_key_state() (tamcolors.ors.tam_tools.tam_key_manager.TAMKeyManager method), 35
 simple_menu_builder() (tamcolors.ors.tam_tools.tam_menu.TAMMenu static method), 36
 slide() (tamcolors.tam_tools.tam_film.TAMFilm method), 33
 slow_test() (in module tamcolors.tests.test_utils), 56
 stability_check() (in module tamcolors.tests.all_tests), 56
 stable() (tamcolors.tam_io.tam_identifier.TAMIdentifier method), 29
 StandardTest (class in tamcolors.tests.tam_io_tests.tam_standard_tests), 43
 start() (tamcolors.tam_io.ansi_256_drivers.ANSI256ColorDriver method), 19
 start() (tamcolors.tam_io.ansi_true_color_drivers.ANSITrueFullColorDriver method), 20
 start() (tamcolors.tam_io.io_tam.IO method), 22
 start() (tamcolors.tam_io.io_tam.RawIO method), 24
 start() (tamcolors.tam_io.tam_drivers.TAMDriver method), 28
 start() (tamcolors.tam_io.win_drivers.WINUtilitiesDriver method), 31
 start_from_bytes() (tamcolors.ors.utils.object_packer.FastHandObjectPacker class method), 58
 step() (tamcolors.tam.tam_loop.TAMLoop method), 18

T

TableTennis (class in tamcolors.ors.examples.tabletennis), 15
 tam_fade_in() (in module tamcolors.ors.tam_tools.tam_fade), 32
 tam_list_buffer() (in module tamcolors.ors.tam_tools.tam_list_buffer), 35
 tam_print() (in module tamcolors.ors.tam_tools.tam_print), 37
 TAMAlpha (class in tamcolors.examples.alpha), 13
 TAMBuffer (class in tamcolors.tam_io.tam_buffer), 24
 TAMBufferTests (class in tamcolors.ors.tests.tam_io_tests.tam_buffer_tests), 40
 TAMButtonRule (class in tamcolors.ors.tam_tools.tam_menu), 35
 TAMColorPalette (class in tamcolors.ors.tam_tools.tam_color_palette), 31
 TAMColorPaletteError, 32
 TAMColorPaletteRule (class in tamcolors.ors.tam_tools.tam_color_palette), 32
 TAMColorPaletteTests (class in tamcolors.ors.tests.tam_tools_tests.tam_color_palette_tests), 45
 TAMCOLORS (class in tamcolors.examples.colors), 14
 tamcolors (module), 60
 tamcolors.examples (module), 17
 tamcolors.examples.alpha (module), 13
 tamcolors.examples.basic (module), 14
 tamcolors.examples.clouds (module), 14
 tamcolors.examples.colors (module), 14
 tamcolors.examples.connection_loopback (module), 14
 tamcolors.examples.host_loopback (module), 14
 tamcolors.examples.icon (module), 14
 tamcolors.examples.rgb_color (module), 15

tamcolors.examples.sandy_check (module), 15
tamcolors.examples.tabletennis (module), 15
tamcolors.examples.tam_key_manager (module), 15
tamcolors.examples.tam_list_buffer (module), 16
tamcolors.examples.tam_loop (module), 16
tamcolors.examples.tam_print (module), 16
tamcolors.examples.tam_text_box (module), 16
tamcolors.tam (module), 18
tamcolors.tam.tam_loop (module), 17
tamcolors.tam_basic (module), 19
tamcolors.tam_basic.basic (module), 19
tamcolors.tam_c (module), 19
tamcolors.tam_io (module), 31
tamcolors.tam_io.ansi_256_drivers (module), 19
tamcolors.tam_io.ansi_true_color_driver (module), 20
tamcolors.tam_io.any_drivers (module), 20
tamcolors.tam_io.io_tam (module), 21
tamcolors.tam_io.tam_buffer (module), 24
tamcolors.tam_io.tam_colors (module), 25
tamcolors.tam_io.tam_drivers (module), 26
tamcolors.tam_io.tam_identifier (module), 29
tamcolors.tam_io.tam_keys (module), 29
tamcolors.tam_io.tcp_io (module), 29
tamcolors.tam_io.uni_drivers (module), 29
tamcolors.tam_io.win_drivers (module), 30
tamcolors.tam_tools (module), 38
tamcolors.tam_tools.tam_color_palette (module), 31
tamcolors.tam_tools.tam_fade (module), 32
tamcolors.tam_tools.tam_film (module), 33
tamcolors.tam_tools.tam_icon (module), 33
tamcolors.tam_tools.tam_key_manager (module), 34
tamcolors.tam_tools.tam_list_buffer (module), 35
tamcolors.tam_tools.tam_menu (module), 35
tamcolors.tam_tools.tam_placing (module), 37
tamcolors.tam_tools.tam_print (module), 37
tamcolors.tam_tools.tam_str (module), 38
tamcolors.tam_tools.tam_text_box (module), 38
tamcolors.tests (module), 56
tamcolors.tests.all_tests (module), 56
tamcolors.tests.tam_basic_tests (module), 39
tamcolors.tests.tam_basic_tests.basic_tests (module), 39
tamcolors.tests.tam_io_tests (module), 44
tamcolors.tests.tam_io_tests.any_drivers_tests (module), 39
tamcolors.tests.tam_io_tests.io_tam_tests (module), 40
tamcolors.tests.tam_io_tests.tam_buffer_tests (module), 40
tamcolors.tests.tam_io_tests.tam_colors_tests (module), 42
tamcolors.tests.tam_io_tests.tam_keys_tests (module), 42
tamcolors.tests.tam_io_tests.tam_standard_tests (module), 43
tamcolors.tests.tam_io_tests.uni_drivers_tests (module), 43
tamcolors.tests.tam_io_tests.win_drivers_tests (module), 43
tamcolors.tests.tam_tests (module), 45
tamcolors.tests.tam_tests.tam_loop_tests (module), 44
tamcolors.tests.tam_tools_tests (module), 52
tamcolors.tests.tam_tools_tests.tam_color_palette_tests (module), 45
tamcolors.tests.tam_tools_tests.tam_fade_tests (module), 46
tamcolors.tests.tam_tools_tests.tam_film_tests (module), 47
tamcolors.tests.tam_tools_tests.tam_key_manager_tests (module), 48
tamcolors.tests.tam_tools_tests.tam_list_buffer_tests (module), 48
tamcolors.tests.tam_tools_tests.tam_menu_tests (module), 49
tamcolors.tests.tam_tools_tests.tam_placing_tests (module), 50
tamcolors.tests.tam_tools_tests.tam_print_tests (module), 51
tamcolors.tests.tam_tools_tests.tam_str_tests (module), 51
tamcolors.tests.tam_tools_tests.tam_text_box_tests (module), 51
tamcolors.tests.test_multi_task_helper (module), 56
tamcolors.tests.test_utils (module), 56
tamcolors.tests.tests_tests (module), 53
tamcolors.tests.tests_tests.test_multi_task_helper (module), 52
tamcolors.tests.utils_tests (module), 56
tamcolors.tests.utils_tests.compress_tests (module), 53
tamcolors.tests.utils_tests.encryption_tests

(module), 53
 tamcolors.tests.utils_tests.identifier_tests (module), 53
 tamcolors.tests.utils_tests.immutable_cache_tests (module), 53
 tamcolors.tests.utils_tests.log_tests (module), 54
 tamcolors.tests.utils_tests.object_packer (module), 54
 tamcolors.tests.utils_tests.tcp_tests (module), 55
 tamcolors.tests.utils_tests.timer_tests (module), 55
 tamcolors.tests.utils_tests.transport_optimizer (module), 55
 tamcolors.utils (module), 60
 tamcolors.utils.compress (module), 57
 tamcolors.utils.encryption (module), 57
 tamcolors.utils.identifier (module), 57
 tamcolors.utils.immutable_cache (module), 57
 tamcolors.utils.log (module), 58
 tamcolors.utils.object_packer (module), 58
 tamcolors.utils.tcp (module), 59
 tamcolors.utils.timer (module), 60
 tamcolors.utils.transport_optimizer (module), 60
 TAMColorTests (class in tamcolors.ors.tests.tam_io_tests.tam_colors_tests), 42
 TAMCycleColor (class in tamcolors.ors.tam_tools.tam_color_palette), 32
 TAMCycleColorTests (class in tamcolors.ors.tests.tam_tools_tests.tam_color_palette_tests), 46
 TAMDefaultColor (class in tamcolors.ors.tam_tools.tam_color_palette), 32
 TAMDefaultColorTests (class in tamcolors.ors.tests.tam_tools_tests.tam_color_palette_tests), 46
 TAMDriver (class in tamcolors.tam_io.tam_drivers), 28
 TAMFilm (class in tamcolors.tam_tools.tam_film), 33
 TAMFilmError, 33
 TAMFilmFadeInTests (class in tamcolors.ors.tests.tam_tools_tests.tam_fade_tests), 46
 TAMFilmTests (class in tamcolors.ors.tests.tam_tools_tests.tam_film_tests), 47
 TAMFrame (class in tamcolors.tam.tam_loop), 17
 TAMFrameTests (class in tamcolors.ors.tests.tam_tests.tam_loop_tests), 44
 TAMIdentifier (class in tamcolors.ors.tam_io.tam_identifier), 29
 TAMKeyManager (class in tamcolors.ors.examples.tam_key_manager), 15
 TAMKeyManager (class in tamcolors.ors.tam_tools.tam_key_manager), 34
 TAMKeyManagerTests (class in tamcolors.ors.tests.tam_tools_tests.tam_key_manager_tests), 48
 TAMKeyTests (class in tamcolors.ors.tests.tam_io_tests.tam_keys_tests), 42
 TAMListBuffer (class in tamcolors.ors.examples.tam_list_buffer), 16
 TAMListBufferTests (class in tamcolors.ors.tests.tam_tools_tests.tam_list_buffer_tests), 48
 TAMLoop (class in tamcolors.tam.tam_loop), 17
 TAMLoopError, 18
 TAMLoopHelloWorld (class in tamcolors.ors.examples.tam_loop), 16
 TAMLoopTests (class in tamcolors.ors.tests.tam_tests.tam_loop_tests), 45
 TAMMenu (class in tamcolors.tam_tools.tam_menu), 35
 TAMMenuTests (class in tamcolors.ors.tests.tam_tools_tests.tam_menu_tests), 49
 TAMPlacingTests (class in tamcolors.ors.tests.tam_tools_tests.tam_placing_tests), 50
 TAMPrint (class in tamcolors.examples.tam_print), 16
 TAMPrint (class in tamcolors.examples.tam_text_box), 16
 TAMPrintTests (class in tamcolors.ors.tests.tam_tools_tests.tam_print_tests), 51
 TAMStrError, 38
 TAMTextBox (class in tamcolors.ors.tam_tools.tam_text_box), 38
 TAMTextBoxButton (class in tamcolors.ors.tam_tools.tam_menu), 36
 TAMTextBoxButtonTests (class in tamcolors.ors.tests.tam_tools_tests.tam_menu_tests), 49
 TAMTextBoxTests (class in tamcolors.ors.tests.tam_tools_tests.tam_text_box_tests), 51
 TAMTextButton (class in tamcolors.ors.tam_tools.tam_menu), 36
 TAMTextButtonTests (class in tamcolors.ors.tests.tam_tools_tests.tam_menu_tests), 50
 task () (tamcolors.tests.test_multi_task_helper.MultiTaskHelper class method), 56
 TCPBase (class in tamcolors.utils.tcp), 59
 TCPConnection (class in tamcolors.utils.tcp), 59
 TCPErrors, 59

TCPHost (class in tamcolors.utils.tcp), 59
 TCPObjectConnector (class in tamcolors.utils.tcp), 59
 TCPObjectWrapper (class in tamcolors.tests.utils_tests.tcp_tests), 55
 TCPObjectWrapper (class in tamcolors.utils.tcp), 59
 TCPReceiver (class in tamcolors.utils.tcp), 60
 TCPTests (class in tamcolors.tests.utils_tests.tcp_tests), 55
 test_1 () (tamcolors.tests.utils_tests.compress_tests.CompressTests method), 53
 test_2 () (tamcolors.tests.utils_tests.compress_tests.CompressTests method), 53
 test_3 () (tamcolors.tests.utils_tests.compress_tests.CompressTests method), 53
 test_4 () (tamcolors.tests.utils_tests.compress_tests.CompressTests method), 53
 test__console_color_count () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 43
 test__done () (tamcolors.tests.tam_tests.tam_loop_tests.TAMFrameTests method), 44
 test__done_2 () (tamcolors.tests.tam_tests.tam_loop_tests.TAMFrameTests method), 44
 test__draw_16 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 43
 test__draw_2 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests static method), 43
 test__draw_onto () (tamcolors.tests.tam_io_tests.io_tam_tests.IOTAMTest method), 40
 test__draw_onto_2 () (tamcolors.tests.tam_io_tests.io_tam_tests.IOTAMTest method), 40
 test__draw_onto_3 () (tamcolors.tests.tam_io_tests.io_tam_tests.IOTAMTest method), 40
 test__get_buffer_dimension () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 43
 test__get_center () (tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests method), 50
 test__get_center_2 () (tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests method), 50
 test__get_color_code_1 () (tamcolors.tests.tam_basic_tests.basic_tests.GetColorCodeTests method), 39
 test__get_color_code_2 () (tamcolors.tests.tam_basic_tests.basic_tests.GetColorCodeTests method), 39
 test__get_color_code_3 () (tamcolors.tests.tam_basic_tests.basic_tests.GetColorCodeTests method), 39
 test__get_color_code_4 () (tamcolors.tests.tam_basic_tests.basic_tests.GetColorCodeTests method), 39
 test__get_dimensions_wrapper () (tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests method), 50
 test__get_dimensions_wrapper_2 () (tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests method), 50
 test__get_dimensions_wrapper_3 () (tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests method), 50
 test__get_other_side () (tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests method), 51
 test__get_other_side_2 () (tamcolors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests method), 51
 test__print () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests static method), 43
 test__print_2 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests static method), 43
 test__print_3 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests static method), 43
 test__processes_special_color_1 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 43
 test__processes_special_color_2 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 43
 test__processes_special_color_3 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 44
 test__processes_special_color_4 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 44
 test__processes_special_color_5 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 44
 test__processes_special_color_6 () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 44
 test__spot_swap () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 44
 test__write_to_output_stream () (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests method), 44

<i>ors.tests.tam_io_tests.io_tam_tests.IOTAMTest</i> <i>static method</i>), 40	<i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test__write_to_output_stream_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.io_tam_tests.IOTAMTest</i> <i>static method</i>), 40	<i>test_buffer_init()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_able_to_execute()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests</i> <i>method</i>), 43	<i>test_buffer_len()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_able_to_execute()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests</i> <i>method</i>), 44	<i>test_buffer_len_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_append()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</i> <i>method</i>), 47	<i>test_buffer_len_3()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_assert_no_main_process()</i> (<i>tamcol-</i> <i>ors.tests.tests_tests.test_multi_task_helper_tests.MultiTaskHelperTests</i> <i>method</i>), 52	<i>test_buffer_len_4()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_assert_no_main_thread()</i> (<i>tamcol-</i> <i>ors.tests.tests_tests.test_multi_task_helper_tests.MultiTaskHelperTests</i> <i>method</i>), 52	<i>test_buffer_ne()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_assert_process()</i> (<i>tamcol-</i> <i>ors.tests.tests_tests.test_multi_task_helper_tests.MultiTaskHelperTests</i> <i>method</i>), 52	<i>test_buffer_ne_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_assert_thread()</i> (<i>tamcol-</i> <i>ors.tests.tests_tests.test_multi_task_helper_tests.MultiTaskHelperTests</i> <i>method</i>), 52	<i>test_buffer_ne_3()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_bool()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPacker.JsonTests</i> <i>method</i>), 54	<i>test_buffer_ne_4()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_bottom_left()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests</i> <i>method</i>), 51	<i>test_buffer_str()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_bottom_left_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests</i> <i>method</i>), 51	<i>test_buffer_str_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40
<i>test_bottom_right()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests</i> <i>method</i>), 51	<i>test_buffer_str_3()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 41
<i>test_bottom_right_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests</i> <i>method</i>), 51	<i>test_build_setting()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.encryption_tests.EncryptionTests</i> <i>method</i>), 53
<i>test_buffer_eq()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40	<i>test_bytearray()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPacker.JsonTests</i> <i>method</i>), 54
<i>test_buffer_eq_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40	<i>test_bytes()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPacker.JsonTests</i> <i>method</i>), 54
<i>test_buffer_eq_3()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40	<i>test_cache()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.immutable_cache_tests.ImmutableCacheTests</i> <i>method</i>), 53
<i>test_buffer_eq_4()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests</i> <i>method</i>), 40	<i>test_call_TextBoxButton()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests</i> <i>static method</i>), 49
<i>test_buffer_eq_5()</i> (<i>tamcol-</i>	<i>test_call_TextButton()</i> (<i>tamcol-</i>

```

    ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests.ors.tests.utils_tests.log_tests.LogTests method),
    static method), 50
    54
test_center() (tamcol- test_debug() (tamcol-
    ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests.ors.tests.utils_tests.log_tests.LogTests method),
    method), 51
    54
test_center_2() (tamcol- test_dict() (tamcol-
    ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests.ors.tests.utils_tests.object_packer_tests.ObjectPacker.JsonTests
    method), 51
    method), 54
test_clear() (tamcol- test_done() (tamcol-
    ors.tests.tam_basic_tests.basic_tests.ClearTests ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests
    static method), 39
    static method), 43
test_clear() (tamcol- test_done() (tamcol-
    ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests
    method), 41
    method), 44
test_clear() (tamcol- test_done() (tamcol-
    ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
    method), 43
    method), 47
test_color_eq() (tamcol- test_done() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.ColorTests ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests
    method), 42
    method), 51
test_color_eq_2() (tamcol- test_done_2() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.ColorTests ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
    method), 42
    method), 47
test_color_eq_3() (tamcol- test_done_2() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.ColorTests ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests
    method), 42
    method), 51
test_color_init() (tamcol- test_done_3() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.ColorTests ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
    method), 42
    method), 47
test_color_init_2() (tamcol- test_draw() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.ColorTests ors.tests.tam_tests.tam_loop_tests.TAMFrameTests
    method), 42
    method), 44
test_color_name_to_color_1() (tamcol- test_draw() (tamcol-
    ors.tests.tam_basic_tests.basic_tests.ColorNameToCodeTests ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
    method), 39
    method), 49
test_color_name_to_color_2() (tamcol- test_draw() (tamcol-
    ors.tests.tam_basic_tests.basic_tests.ColorNameToCodeTests ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
    method), 39
    method), 49
test_color_ne() (tamcol- test_draw() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.ColorTests ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
    method), 42
    method), 50
test_color_ne_2() (tamcol- test_draw() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.ColorTests ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests
    method), 42
    method), 52
test_colors() (tamcol- test_draw_2() (tamcol-
    ors.tests.tam_io_tests.tam_colors_tests.TAMColorTests ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
    method), 42
    method), 49
test_copy() (tamcol- test_draw_2() (tamcol-
    ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
    method), 41
    method), 50
test_copy_2() (tamcol- test_draw_2() (tamcol-
    ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests
    method), 41
    method), 52
test_critical() (tamcol- test_draw_3() (tamcol-

```

ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests
 method), 52

ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
 method), 47

test_draw_onto() (tamcol- test_get_3() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests*
 method), 41 method), 47

test_draw_onto_2() (tamcol- test_get_action() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests*
 method), 41 method), 49

test_draw_onto_3() (tamcol- test_get_action() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests*
 method), 41 method), 50

test_draw_onto_4() (tamcol- test_get_buttons() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests*
 method), 41 method), 49

test_draw_onto_5() (tamcol- test_get_buttons_2() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests*
 method), 41 method), 49

test_draw_onto_6() (tamcol- test_get_buttons_3() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests*
 method), 41 method), 49

test_draw_onto_7() (tamcol- test_get_call_key() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests*
 method), 41 method), 49

test_draw_onto_8() (tamcol- test_get_call_key_2() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests*
 method), 41 method), 49

test_dunder_bytes() (tamcol- test_get_char() (tamcol-
ors.tests.utils_tests.object_packer_tests.FastHandObjectPackerTests *ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests*
 method), 54 method), 52

test_enable_and_disable_logging() (tam- test_get_char_2() (tamcol-
colors.tests.utils_tests.log_tests.LogTests static *ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests*
 method), 54 method), 52

test_encrypt_decrypt() (tamcol- test_get_circular() (tamcol-
ors.tests.utils_tests.encryption_tests.EncryptionTests *ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests*
 method), 53 method), 47

test_fast_hand_object() (tamcol- test_get_circular_2() (tamcol-
ors.tests.utils_tests.object_packer_tests.ObjectPacker.JsonTests *ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests*
 method), 54 method), 47

test_float() (tamcol- test_get_clock() (tamcol-
ors.tests.utils_tests.object_packer_tests.ObjectPacker.JsonTests *ors.tests.tam_tools_tests.tam_color_palette_tests.TAMCycleColor*
 method), 54 method), 46

test_flush_and_stderr() (tamcol- test_get_color() (tamcol-
ors.tests.tam_basic_tests.basic_tests.PrintCTests *ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet*
 method), 39 method), 45

test_frame_init() (tamcol- test_get_color() (tamcol-
ors.tests.tam_tests.tam_loop_tests.TAMFrameTests *ors.tests.tam_tools_tests.tam_color_palette_tests.TAMDefaultCol*
 method), 44 method), 46

test_generate_identifier() (tamcol- test_get_color_2() (tamcol-
ors.tests.utils_tests.identifier_tests.IdentifierTests *ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet*
 method), 53 method), 45

test_get() (tamcol- test_get_color_3() (tamcol-
ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests *ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet*
 method), 47 method), 45

test_get_2() (tamcol- test_get_colors() (tamcol-

ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
method), 52 *method*), 49

test_get_colors_2() (tamcol- test_get_height_min_and_max() (tamcol-
ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests.tests.tam_tests.tam_loop_tests.TAMFrameTests
method), 52 *method*), 45

test_get_cross_rect() (tamcol- test_get_height_min_and_max_2() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_tests.tam_loop_tests.TAMFrameTests*
method), 41 *method*), 45

test_get_cross_rect_2() (tamcol- test_get_identifier_bytes() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.utils_tests.identifier_tests.IdentifierTests*
method), 41 *method*), 53

test_get_defaults() (tamcol- test_get_io() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_io_tests.any_drivers_tests.AnyIOTests*
method), 41 *method*), 39

test_get_defaults() (tamcol- test_get_key() (tamcol-
ors.tests.tam_tests.tam_loop_tests.TAMFrameTests *ors.tests.tam_io_tests.any_drivers_tests.AnyIOTests*
method), 44 *method*), 39

test_get_defaults_2() (tamcol- test_get_key() (tamcol-
ors.tests.tam_tests.tam_loop_tests.TAMFrameTests *ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests*
method), 45 *method*), 43

test_get_dimensions() (tamcol- test_get_key() (tamcol-
ors.tests.tam_io_tests.any_drivers_tests.AnyIOTests *ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests*
method), 39 *method*), 44

test_get_dimensions() (tamcol- test_get_key_2() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests*
method), 41 *method*), 43

test_get_dimensions() (tamcol- test_get_key_2() (tamcol-
ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests *ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests*
method), 43 *method*), 44

test_get_dimensions() (tamcol- test_get_key_3() (tamcol-
ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests *ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests*
method), 44 *method*), 43

test_get_dimensions() (tamcol- test_get_key_3() (tamcol-
ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests.tests.tam_io_tests.win_drivers_tests.WinDriversTests
method), 52 *method*), 44

test_get_dimensions_2() (tamcol- test_get_key_4() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests*
method), 41 *method*), 43

test_get_dimensions_2() (tamcol- test_get_key_4() (tamcol-
ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests.tests.tam_io_tests.win_drivers_tests.WinDriversTests
method), 52 *method*), 44

test_get_dimensions_3() (tamcol- test_get_key_5() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests*
method), 41 *method*), 43

test_get_fps() (tamcol- test_get_key_5() (tamcol-
ors.tests.tam_tests.tam_loop_tests.TAMFrameTests *ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests*
method), 45 *method*), 44

test_get_fps_2() (tamcol- test_get_key_dict() (tamcol-
ors.tests.tam_tests.tam_loop_tests.TAMFrameTests *ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests*
method), 45 *method*), 43

test_get_from_raw_spot() (tamcol- test_get_key_dict() (tamcol-
ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests *ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests*
method), 41 *method*), 44

test_get_goto_map() (tamcol- test_get_key_state() (tamcol-

`ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 48`

`test_get_key_state_2()` (tamcol- `test_get_rule()` (tamcol-
`ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 48`

`test_get_modes()` (tamcol- `test_get_rule_2()` (tamcol-
`ors.tests.tam_io_tests.any_drivers_tests.AnyIOTests` `ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests`
`method), 39` `method), 45`

`test_get_modes()` (tamcol- `test_get_spot()` (tamcol-
`ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests` `ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests`
`method), 43` `method), 41`

`test_get_modes()` (tamcol- `test_get_text()` (tamcol-
`ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests` `ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests`
`method), 44` `method), 52`

`test_get_on()` (tamcol- `test_get_text_2()` (tamcol-
`ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests` `ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests`
`method), 49` `method), 52`

`test_get_on_2()` (tamcol- `test_get_user_input()` (tamcol-
`ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests` `ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 49` `method), 48`

`test_get_position()` (tamcol- `test_get_user_input_2()` (tamcol-
`ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests` `ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 49` `method), 48`

`test_get_position()` (tamcol- `test_get_user_input_3()` (tamcol-
`ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests` `ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 50` `method), 48`

`test_get_position_2()` (tamcol- `test_get_user_input_generator()` (tamcol-
`ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests` `ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 49` `method), 48`

`test_get_position_2()` (tamcol- `test_get_user_input_generator_2()` (tamcol-
`ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests` `ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 50` `method), 48`

`test_get_property()` (tamcol- `test_get_user_input_generator_3()` (tamcol-
`ors.tests.tam_io_tests.tam_colors_tests.ColorTests` `ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests`
`method), 42` `method), 48`

`test_get_property()` (tamcol- `test_get_width_min_and_max()` (tamcol-
`ors.tests.tam_io_tests.tam_colors_tests.RGBATests` `ors.tests.tam_tests.tam_loop_tests.TAMFrameTests`
`method), 42` `method), 45`

`test_get_property_2()` (tamcol- `test_get_width_min_and_max_2()` (tamcol-
`ors.tests.tam_io_tests.tam_colors_tests.ColorTests` `ors.tests.tam_tests.tam_loop_tests.TAMFrameTests`
`method), 42` `method), 45`

`test_get_property_2()` (tamcol- `test_getitem()` (tamcol-
`ors.tests.tam_io_tests.tam_colors_tests.RGBATests` `ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests`
`method), 42` `method), 45`

`test_get_raw_buffers()` (tamcol- `test_getitem()` (tamcol-
`ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests` `ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests`
`method), 41` `method), 47`

`test_get_raw_buffers_2()` (tamcol- `test_getitem_2()` (tamcol-
`ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests` `ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests`
`method), 41` `method), 46`

`test_get_raw_spot()` (tamcol- `test_getitem_2()` (tamcol-
`ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests` `ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests`
`method), 41` `method), 47`

`test_get_raw_user_input()` (tamcol- `test_getitem_3()` (tamcol-

<code>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests</code> <code>method</code>), 46	<code>ors.tests.tam_basic_tests.basic_tests.InputCTests</code> <code>method</code>), 39
<code>test_getitem_3()</code> <code>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</code> <code>method</code>), 47	<code>test_int()</code> <code>ors.tests.utils_tests.object_packer_tests.ObjectPackerJsonTests</code> <code>method</code>), 54
<code>test_globals()</code> <code>ors.tests.utils_tests.identifier_tests.IdentifierTests</code> <code>method</code>), 53	<code>test_IO()</code> (<code>tamcolors.tests.tam_basic_tests.basic_tests.IOTests</code> <code>method</code>), 39
<code>test_hand_down()</code> <code>ors.tests.tests_tests.test_multi_task_helper_tests.MultiTaskHelperTests</code> <code>method</code>), 52	<code>test_iter()</code> <code>ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests</code> <code>method</code>), 48
<code>test_hide_console_cursor()</code> <code>ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests</code> <code>static method</code>), 43	<code>test_key_present()</code> <code>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests</code> <code>method</code>), 46
<code>test_init()</code> <code>ors.tests.utils_tests.encryption_tests.EncryptionTests</code> <code>method</code>), 53	<code>test_key_present_2()</code> <code>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests</code> <code>method</code>), 46
<code>test_init()</code> <code>ors.tests.utils_tests.encryption_tests.NoEncryptionTests</code> <code>method</code>), 53	<code>test_keys()</code> <code>ors.tests.tam_io_tests.tam_keys_tests.TAMKeyTests</code> <code>method</code>), 42
<code>test_init()</code> <code>ors.tests.utils_tests.immutable_cache_tests.ImmutableCacheTests</code> <code>method</code>), 54	<code>test_lap()</code> <code>ors.tests.utils_tests.timer_tests.TimerTests</code> <code>method</code>), 55
<code>test_init_color_palette()</code> <code>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests</code> <code>method</code>), 46	<code>test_large_data()</code> <code>ors.tests.utils_tests.object_packer_tests.ObjectPackerJsonTests</code> <code>method</code>), 55
<code>test_init_color_palette_2()</code> <code>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests</code> <code>method</code>), 46	<code>test_large_data()</code> <code>ors.tests.utils_tests.transport_optimizer_tests.TransportOptimizerTests</code> <code>method</code>), 55
<code>test_init_cycle_color()</code> <code>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMCycleColorTests</code> <code>method</code>), 46	<code>test_large_tam_buffer_data()</code> <code>ors.tests.utils_tests.transport_optimizer_tests.TransportOptimizerTests</code> <code>method</code>), 55
<code>test_init_default_color()</code> <code>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMDefaultColorTests</code> <code>method</code>), 46	<code>test_len()</code> <code>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</code> <code>method</code>), 47
<code>test_init_film()</code> <code>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</code> <code>method</code>), 47	<code>test_len_2()</code> <code>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</code> <code>method</code>), 47
<code>test_init_key_manger()</code> <code>ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests</code> <code>method</code>), 48	<code>test_list()</code> <code>ors.tests.utils_tests.object_packer_tests.ObjectPackerJsonTests</code> <code>method</code>), 54
<code>test_init_TextBoxButton()</code> <code>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests</code> <code>method</code>), 49	<code>test_loop_init()</code> <code>ors.tests.tam_tests.tam_loop_tests.TAMLoopTests</code> <code>method</code>), 45
<code>test_init_TextBoxButton_2()</code> <code>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests</code> <code>method</code>), 49	<code>test_make_buffer_ready()</code> <code>ors.tests.tam_tests.tam_loop_tests.TAMFrameTests</code> <code>method</code>), 45
<code>test_init_TextButton()</code> <code>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests</code> <code>method</code>), 50	<code>test_make_buffer_ready_2()</code> <code>ors.tests.tam_tests.tam_loop_tests.TAMFrameTests</code> <code>method</code>), 45
<code>test_init_TextButton_2()</code> <code>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests</code> <code>method</code>), 50	<code>test_make_buffer_ready_3()</code> <code>ors.tests.tam_tests.tam_loop_tests.TAMFrameTests</code> <code>method</code>), 45
<code>test_inputc()</code> <code>ors.tests.tam_tools_tests.tam_str_tests.MakeTAMStrTests</code> <code>method</code>), 49	<code>test_make_tam_str()</code> <code>ors.tests.tam_tools_tests.tam_str_tests.MakeTAMStrTests</code> <code>method</code>), 49

method), 51
 test_make_tam_str_2() (tamcol- ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
 method), 50
 ors.tests.tam_tools_tests.tam_str_tests.MakeTAMStrTests.peak() (tamcol-
 method), 51
 ors.tests.tam_tools_tests.tam_str_tests.MakeTAMStrTests.peak_2() (tamcol-
 method), 51
 ors.tests.tam_tools_tests.tam_str_tests.MakeTAMStrTests.peak_3() (tamcol-
 method), 51
 ors.tests.tam_tools_tests.tam_str_tests.MakeTAMStrTests.spinning() (tamcol-
 method), 51
 test_new() (tamcol- ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
 method), 55
 ors.tests.utils_tests.immutable_cache_tests.ImmutableCacheTests.ipv6() (tamcol-
 method), 54
 ors.tests.utils_tests.tcp_tests.TCPTests
 test_next() (tamcol- method), 55
 ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests.test_pop() (tamcol-
 method), 47
 ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
 test_next_2() (tamcol- method), 47
 ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests.test_preferred_mode() (tamcol-
 method), 47
 ors.tests.tam_tests.tam_loop_tests.TAMLoopTests
 test_none() (tamcol- method), 45
 ors.tests.utils_tests.object_packer_tests.ObjectPackerTests.test_preferred_mode_2() (tamcol-
 method), 55
 ors.tests.tam_tests.tam_loop_tests.TAMLoopTests
 test_not_same_color() (tamcol- method), 45
 ors.tests.tam_basic_tests.basic_tests.PrintCTests test_preferred_mode_3() (tamcol-
 method), 39
 ors.tests.tam_tests.tam_loop_tests.TAMLoopTests
 test_object_wrapper() (tamcol- method), 45
 ors.tests.utils_tests.tcp_tests.TCPObjectWrapper test_replace_alpha_chars_1() (tamcol-
 method), 55
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 test_off() (tamcol- method), 41
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests.test_replace_alpha_chars_2() (tamcol-
 method), 49
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 test_off() (tamcol- method), 41
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests.test_replace_alpha_chars_3() (tamcol-
 method), 50
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 test_off_2() (tamcol- method), 41
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests.test_reset_colors_to_console_defaults() (tamcol-
 method), 49
 (tamcolors.tests.tam_io_tests.any_drivers_tests.AnyIOTests
 static method), 39
 test_off_2() (tamcol- method), 41
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests.test_reset_colors_to_console_defaults() (tamcol-
 method), 50
 (tamcolors.tests.tam_io_tests.win_drivers_tests.WinDriversTests
 method), 44
 test_offset_sleep() (tamcol- method), 44
 ors.tests.utils_tests.timer_tests.TimerTests test_reset_colors_to_console_defaults() (tamcol-
 method), 55
 (tamcolors.tests.tam_tests.tam_loop_tests.TAMLoopTests
 method), 49
 (tamcolors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
 method), 49
 test_on() (tamcolors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
 method), 50
 test_on() (tamcolors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
 method), 50
 test_on_2() (tamcol- test_rgba_eq_2() (tamcol-
 method), 49
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests.test_rgba_eq_2() (tamcol-
 method), 42
 (tamcolors.tests.tam_io_tests.tam_colors_tests.RGBATests
 method), 42
 test_on_2() (tamcol- test_rgba_eq_3() (tamcol-
 method), 42
 (tamcolors.tests.tam_io_tests.tam_colors_tests.RGBATests
 method), 42

<i>ors.tests.tam_io_tests.tam_colors_tests.RGBATests</i> <i>method</i>), 42	<i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTe</i> <i>method</i>), 54
<i>test_rgba_init()</i> (<i>tamcol-</i> <i>test_save_and_load_int_3()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_colors_tests.RGBATests</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTe</i> <i>method</i>), 42 <i>method</i>), 54	
<i>test_rgba_init_2()</i> (<i>tamcol-</i> <i>test_save_and_load_int_4()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_colors_tests.RGBATests</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTe</i> <i>method</i>), 42 <i>method</i>), 54	
<i>test_rgba_ne()</i> (<i>tamcol-</i> <i>test_set()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_colors_tests.RGBATests</i> <i>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</i> <i>method</i>), 42 <i>method</i>), 47	
<i>test_rgba_ne_2()</i> (<i>tamcol-</i> <i>test_set()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_colors_tests.RGBATests</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPacker.JsonTests</i> <i>method</i>), 42 <i>method</i>), 55	
<i>test_run()</i> (<i>tamcol-</i> <i>test_set_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_tests.tam_loop_tests.TAMLoopTests</i> <i>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</i> <i>method</i>), 45 <i>method</i>), 47	
<i>test_run_action()</i> (<i>tamcol-</i> <i>test_set_3()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests</i> <i>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</i> <i>static method</i>), 49 <i>method</i>), 47	
<i>test_run_action()</i> (<i>tamcol-</i> <i>test_set_action()</i> (<i>tamcol-</i> <i>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests</i> <i>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests</i> <i>static method</i>), 50 <i>method</i>), 50	
<i>test_same_color()</i> (<i>tamcol-</i> <i>test_set_action()</i> (<i>tamcol-</i> <i>ors.tests.tam_basic_tests.basic_tests.PrintCTests</i> <i>ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests</i> <i>static method</i>), 39 <i>method</i>), 50	
<i>test_same_data()</i> (<i>tamcol-</i> <i>test_set_char()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.transport_optimizer_tests.TransportOptimizerTests</i> <i>ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests</i> <i>method</i>), 55 <i>method</i>), 52	
<i>test_same_doc_string()</i> (<i>tamcol-</i> <i>test_set_char_2()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.io_tam_tests.RAWIOTest</i> <i>ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests</i> <i>method</i>), 40 <i>method</i>), 52	
<i>test_same_keys()</i> (<i>tamcol-</i> <i>test_set_circular()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.tam_standard_tests.StandardTest</i> <i>ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests</i> <i>method</i>), 43 <i>method</i>), 47	
<i>test_same_methods()</i> (<i>tamcol-</i> <i>test_set_clock()</i> (<i>tamcol-</i> <i>ors.tests.tam_io_tests.io_tam_tests.RAWIOTest</i> <i>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMCycleColor</i> <i>method</i>), 40 <i>method</i>), 46	
<i>test_save_and_load_data_1()</i> (<i>tamcol-</i> <i>test_set_color()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTests</i> <i>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet</i> <i>method</i>), 54 <i>method</i>), 46	
<i>test_save_and_load_data_2()</i> (<i>tamcol-</i> <i>test_set_color()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTests</i> <i>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMDefaultCol</i> <i>method</i>), 54 <i>method</i>), 46	
<i>test_save_and_load_data_3()</i> (<i>tamcol-</i> <i>test_set_color_2()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTests</i> <i>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet</i> <i>method</i>), 54 <i>method</i>), 46	
<i>test_save_and_load_data_4()</i> (<i>tamcol-</i> <i>test_set_color_3()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTests</i> <i>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet</i> <i>method</i>), 54 <i>method</i>), 46	
<i>test_save_and_load_int_1()</i> (<i>tamcol-</i> <i>test_set_colors()</i> (<i>tamcol-</i> <i>ors.tests.utils_tests.object_packer_tests.ObjectPackerFunctionsTests</i> <i>ors.tests.tam_tools_tests.tam_color_palette_tests.TAMCycleColor</i> <i>method</i>), 54 <i>method</i>), 46	
<i>test_save_and_load_int_2()</i> (<i>tamcol-</i> <i>test_set_colors()</i> (<i>tamcol-</i>	

ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests.ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 method), 52 method), 41
 test_set_colors_2() (tamcol- test_set_spot_4() (tamcol-
 ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests.ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 method), 52 method), 41
 test_set_colors_3() (tamcol- test_set_spot_5() (tamcol-
 ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests.ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 method), 52 method), 41
 test_set_defaults_and_clear() (tamcol- test_set_spot_6() (tamcol-
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 method), 41 method), 41
 test_set_defaults_and_clear_2() (tamcol- test_set_spot_7() (tamcol-
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 method), 41 method), 42
 test_set_dimensions_and_clear() (tamcol- test_set_spot_8() (tamcol-
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 method), 41 method), 42
 test_set_dimensions_and_clear_2() (tam- test_set_spot_9() (tamcol-
 colors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests
 method), 41 method), 42
 test_set_position() (tamcol- test_set_tam_color_defaults() (tamcol-
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests.ors.tests.tam_io_tests.any_drivers_tests.AnyIOTests
 method), 50 static method), 40
 test_set_position() (tamcol- test_set_tam_color_defaults() (tamcol-
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests.ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests
 method), 50 method), 44
 test_set_position_2() (tamcol- test_set_tam_color_defaults() (tamcol-
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests.ors.tests.tam_tests.tam_loop_tests.TAMLoopTests
 method), 50 method), 45
 test_set_position_2() (tamcol- test_setitem() (tamcol-
 ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests.ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet
 method), 50 method), 46
 test_set_rule() (tamcol- test_setitem() (tamcol-
 ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletTests.ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
 method), 46 method), 47
 test_set_rule_2() (tamcol- test_setitem_2() (tamcol-
 ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletTests.ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet
 method), 46 method), 46
 test_set_slash_get_mode() (tamcol- test_setitem_2() (tamcol-
 ors.tests.tam_io_tests.any_drivers_tests.AnyIOTests ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
 method), 40 method), 47
 test_set_slash_get_mode() (tamcol- test_setitem_3() (tamcol-
 ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPalet
 method), 43 method), 46
 test_set_slash_get_mode() (tamcol- test_setitem_3() (tamcol-
 ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
 method), 44 method), 47
 test_set_spot() (tamcol- test_show_console_cursor() (tamcol-
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests
 method), 41 static method), 43
 test_set_spot_2() (tamcol- test_silent_key_state() (tamcol-
 ors.tests.tam_io_tests.tam_buffer_tests.TAMBufferTests ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManag
 method), 41 method), 48
 test_set_spot_3() (tamcol- test_silent_key_state_2() (tamcol-

```

    ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests
    method), 48
    ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests
    method), 50
test_silent_key_state_3() (tamcol- test_str_TextBoxButton_2() (tamcol-
    ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests
    method), 48
    ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManagerTests
    method), 50
test_simple_data() (tamcol- test_str_TextButton() (tamcol-
    ors.tests.utils_tests.transport_optimizer_tests.TransportOptimizerTests
    method), 55
    ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
    method), 50
test_simple_encrypt_decrypt() (tamcol- test_str_TextButton_2() (tamcol-
    ors.tests.utils_tests.encryption_tests.EncryptionTests
    method), 53
    ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
    method), 50
test_simple_menu_builder() (tamcol- test_tam_buffer_1() (tamcol-
    ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
    method), 49
    ors.tests.utils_tests.object_packer_tests.FastHandObjectPackerTests
    method), 54
test_simple_menu_builder_2() (tamcol- test_tam_buffer_2() (tamcol-
    ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
    method), 49
    ors.tests.utils_tests.object_packer_tests.FastHandObjectPackerTests
    method), 54
test_simple_menu_builder_3() (tamcol- test_tam_color_1() (tamcol-
    ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
    method), 49
    ors.tests.utils_tests.object_packer_tests.FastHandObjectPackerTests
    method), 54
test_simple_menu_builder_4() (tamcol- test_tam_color_2() (tamcol-
    ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
    method), 49
    ors.tests.utils_tests.object_packer_tests.FastHandObjectPackerTests
    method), 54
test_simple_menu_builder_5() (tamcol- test_tam_fade_in() (tamcol-
    ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
    method), 49
    ors.tests.tam_tools_tests.tam_fade_tests.TAMFilmFadeInTests
    method), 46
test_slide() (tamcol- test_tam_fade_in_2() (tamcol-
    ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
    method), 47
    ors.tests.tam_tools_tests.tam_fade_tests.TAMFilmFadeInTests
    method), 46
test_slide_2() (tamcol- test_tam_fade_in_3() (tamcol-
    ors.tests.tam_tools_tests.tam_film_tests.TAMFilmTests
    method), 47
    ors.tests.tam_tools_tests.tam_fade_tests.TAMFilmFadeInTests
    method), 47
test_stack() (tamcol- test_tam_fade_in_4() (tamcol-
    ors.tests.tam_tests.tam_loop_tests.TAMLoopTests
    method), 45
    ors.tests.tam_tools_tests.tam_fade_tests.TAMFilmFadeInTests
    method), 47
test_start() (tamcol- test_tam_list_buffer() (tamcol-
    ors.tests.tam_io_tests.uni_drivers_tests.UniIOTests
    static method), 43
    ors.tests.tam_tools_tests.tam_list_buffer_tests.TAMListBufferTests
    method), 48
test_start() (tamcol- test_tam_list_buffer_2() (tamcol-
    ors.tests.tam_io_tests.win_drivers_tests.WinDriversTests
    method), 44
    ors.tests.tam_tools_tests.tam_list_buffer_tests.TAMListBufferTests
    method), 48
test_step() (tamcol- test_tam_list_buffer_3() (tamcol-
    ors.tests.tam_tests.tam_loop_tests.TAMLoopTests
    method), 45
    ors.tests.tam_tools_tests.tam_list_buffer_tests.TAMListBufferTests
    method), 48
test_str() (tamcol- test_tam_list_buffer_4() (tamcol-
    ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests
    method), 46
    ors.tests.tam_tools_tests.tam_list_buffer_tests.TAMListBufferTests
    method), 48
test_str() (tamcol- test_tam_list_buffer_5() (tamcol-
    ors.tests.utils_tests.object_packer_tests.ObjectPackerJsonTests
    method), 55
    ors.tests.tam_tools_tests.tam_list_buffer_tests.TAMListBufferTests
    method), 48
test_str_2() (tamcol- test_tam_menu() (tamcol-
    ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests
    method), 46
    ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
    method), 49
test_str_TextBoxButton() (tamcol- test_tam_menu_2() (tamcol-

```

ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
 method), 49

test_tam_print() (tamcol- test_update() (tamcol-
 ors.tests.tam_tools_tests.tam_print_tests.TAMPrintTests ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManag
 method), 51 method), 48

test_tam_print_2() (tamcol- test_update() (tamcol-
 ors.tests.tam_tools_tests.tam_print_tests.TAMPrintTests ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
 method), 51 method), 49

test_tam_print_3() (tamcol- test_update() (tamcol-
 ors.tests.tam_tools_tests.tam_print_tests.TAMPrintTests ors.tests.tam_tools_tests.tam_menu_tests.TAMTextBoxButtonTests
 method), 51 static method), 50

test_tam_print_4() (tamcol- test_update() (tamcol-
 ors.tests.tam_tools_tests.tam_print_tests.TAMPrintTests ors.tests.tam_tools_tests.tam_menu_tests.TAMTextButtonTests
 method), 51 static method), 50

test_tam_print_5() (tamcol- test_update() (tamcol-
 ors.tests.tam_tools_tests.tam_print_tests.TAMPrintTests ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests
 method), 51 method), 52

test_tam_text_box_init() (tamcol- test_update_2() (tamcol-
 ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests ors.tests.tam_tools_tests.tam_color_palette_tests.TAMCycleColor
 method), 52 method), 46

test_tam_text_box_str() (tamcol- test_update_2() (tamcol-
 ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests ors.tests.tam_tools_tests.tam_key_manager_tests.TAMKeyManag
 method), 52 method), 48

test_tam_text_box_str_2() (tamcol- test_update_2() (tamcol-
 ors.tests.tam_tools_tests.tam_text_box_tests.TAMTextBoxTests ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
 method), 52 method), 49

test_timer() (tamcol- test_update_3() (tamcol-
 ors.tests.utils_tests.timer_tests.TimerTests ors.tests.tam_tools_tests.tam_menu_tests.TAMMenuTests
 method), 55 method), 49

test_top_left() (tamcol- test_win_stable() (tamcol-
 ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests ors.tests.tam_io_tests.win_drivers_tests.WinGlobalsTests
 method), 51 method), 44

test_top_left_2() (tamcol- tests_main() (in module tamcolors.tests.all_tests),
 ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests 56
 method), 51 Timer (class in tamcolors.utils.timer), 60

test_top_right() (tamcol- TimerTests (class in tamcol-
 ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests ors.tests.utils_tests.timer_tests), 55
 method), 51 to_bytes() (tamcolors.tam_io.tam_buffer.TAMBuffer

test_top_right_2() (tamcol- method), 25
 ors.tests.tam_tools_tests.tam_placing_tests.TAMPlacingTests (tamcolors.tam_io.tam_colors.Color
 method), 51 method), 26

test_tuple() (tamcol- to_bytes() (tamcolors.tam_io.tam_colors.RGBA
 ors.tests.utils_tests.object_packer_tests.ObjectPackerJsonTests method), 26
 method), 55 to_bytes() (tamcol-

test_update() (tamcol- ors.utils.object_packer.FastHandObjectPacker
 ors.tests.tam_tests.tam_loop_tests.TAMFrameTests method), 58
 method), 45 top_left() (in module tamcol-

test_update() (tamcol- ors.tam_tools.tam_placing), 37
 ors.tests.tam_tools_tests.tam_color_palette_tests.TAMColorPaletteTests (in module tamcol-
 method), 46 ors.tam_tools.tam_placing), 37

test_update() (tamcol- transparent_value (tamcol-
 ors.tests.tam_tools_tests.tam_color_palette_tests.TAMCycleColorTests ors.tam_colors.Color attribute),
 method), 46 26

test_update() (tamcol- TransportOptimizerTests (class in tamcol-

ors.tests.utils_tests.transport_optimizer_tests),
55

U

UniIOTests (class in *tamcol-
ors.tests.tam_io_tests.uni_drivers_tests*),
43

UNIKeyDriver (class in *tamcol-
ors.tam_io.uni_drivers*), 29

UNISharedData (class in *tamcol-
ors.tam_io.uni_drivers*), 29

UNIUtilitiesDriver (class in *tamcol-
ors.tam_io.uni_drivers*), 30

update() (*tamcolors.examples.alpha.TAMAlpha*
method), 13

update() (*tamcolors.examples.clouds.Clouds* *method*),
14

update() (*tamcolors.examples.colors.TAMCOLORS*
method), 14

update() (*tamcolors.examples.icon.BootLogo*
method), 14

update() (*tamcolors.examples.rgb_color.RGBCOLOR*
method), 15

update() (*tamcolors.examples.tabletennis.Ball*
method), 15

update() (*tamcolors.examples.tabletennis.Racket*
method), 15

update() (*tamcolors.examples.tabletennis.TableTennis*
method), 15

update() (*tamcolors.examples.tam_key_manager.TAMKeyManager*
method), 15

update() (*tamcolors.examples.tam_list_buffer.TAMListBuffer*
method), 16

update() (*tamcolors.examples.tam_loop.TAMLoopHelloWorld*
method), 16

update() (*tamcolors.examples.tam_print.TAMPrint*
method), 16

update() (*tamcolors.examples.tam_text_box.TAMPrint*
method), 16

update() (*tamcolors.tam.tam_loop.TAMFrame*
method), 17

update() (*tamcolors.tam_tools.tam_color_palette.TAMColorPalette*
method), 32

update() (*tamcolors.tam_tools.tam_color_palette.TAMColorPaletteRule*
method), 32

update() (*tamcolors.tam_tools.tam_color_palette.TAMCycleColor*
method), 32

update() (*tamcolors.tam_tools.tam_color_palette.TAMDefaultColor*
method), 32

update() (*tamcolors.tam_tools.tam_key_manager.TAMKeyManager*
method), 35

update() (*tamcolors.tam_tools.tam_menu.TAMButtonRule*
method), 35

update() (*tamcolors.tam_tools.tam_menu.TAMMenu*
method), 36

update() (*tamcolors.tam_tools.tam_menu.TAMTextBoxButton*
method), 36

update() (*tamcolors.tam_tools.tam_menu.TAMTextButton*
method), 37

update() (*tamcolors.tam_tools.tam_text_box.TAMTextBox*
method), 38

utilities_driver_operational() (*tamcol-
ors.tam_io.io_tam.IO* *method*), 22

utilities_driver_operational() (*tamcol-
ors.tam_io.io_tam.RawIO* *method*), 24

UtilitiesDriver (class in *tamcol-
ors.tam_io.tam_drivers*), 28

W

warning() (in module *tamcolors.utils.log*), 58

WinDriversTests (class in *tamcol-
ors.tests.tam_io_tests.win_drivers_tests*),
43

WINFullColorDriver (class in *tamcol-
ors.tam_io.win_drivers*), 30

WinGlobalsTests (class in *tamcol-
ors.tests.tam_io_tests.win_drivers_tests*),
44

WINKeyDriver (class in *tamcol-
ors.tam_io.win_drivers*), 30

winner() (*tamcolors.examples.tabletennis.Ball*
method), 15

WinSharedData (class in *tamcol-
ors.tam_io.win_drivers*), 31

WinUtilitiesDriver (class in *tamcol-
ors.tam_io.win_drivers*), 31